EXHIBIT C

REDACTED VERSION OF DOCUMENT SOUGHT TO BE SEALED

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISIONS

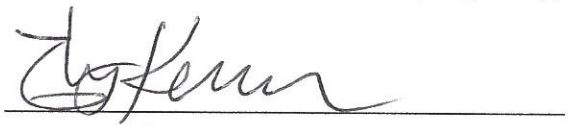
ORACLE AMERICA, INC.,	)	
	)	
Plaintiff,	)	
	)	
v.	)	Civil Action No. 10-03561 WHA
	)	
GOOGLE INC.,	)	
	)	
Defendant.	)	

EXPERT REPORT OF CHRIS F. KEMERER, Ph.D.

January 8, 2016

CONFIDENTIAL – ATTORNEYS' EYES ONLY
PURSUANT TO PROTECTIVE ORDER

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct. Executed on this 8th day of January, 2016, in Pittsburgh, PA.



Chris F. Kemerer, PhD
Chris F. Kemerer PhD LLC

TABLE OF CONTENTS

.

I. Assignment

1. I have been asked to provide an overview and opinion concerning the historical origins, evolution, and adoption of the declaring code and structure, sequence and organization of the 37 specific Java Application Programming Interface packages ("Java APIs") as they relate to the development and proliferation of Google's Android platform. I have also been asked to examine whether there is copying of the code and structure, sequence and organization of the Java APIs in Android. I have also been asked to evaluate the nature and purpose of both the Java APIs and the Android system that draws on them, the potential benefits to Google from its commercial use of the Java APIs in Android, and potential injuries that Google's use of the Java APIs causes to the Java platform and ecosystem.

2. In addressing these questions, I draw on my academic and professional background, which includes more than 30 years of experience in the areas of software engineering, technology adoption and diffusion, the creation and management of information systems, and the market economics of software and information systems.

3. I am being compensated for my work on this case at a rate of $595 per hour. My compensation is not contingent upon my testimony or on the result of this proceeding. Appendix C lists the materials I considered in preparing this report.

4. My work is ongoing, and it is my understanding that discovery in this case is ongoing. Accordingly, I reserve the right to supplement or amend my opinions in light of any additional evidence, testimony, or information that may be provided to me after the date of this report. I also reserve the right to supplement or amend my opinions in response to any expert reports served by any party in the lawsuit.

II. Qualifications

5. I hold a Doctor of Philosophy (PhD) degree in (Information) Systems Sciences from the Graduate School of Industrial Administration at Carnegie Mellon University. I also hold a Master's Degree from Carnegie Mellon as well as a Bachelor's Degree in Decision Sciences and Economics from the Wharton School at the University of Pennsylvania.

6. Prior to my graduate studies I managed commercial software development projects in industry for a variety of public and private sector clients. I have also conducted and supervised a significant number of research projects involving the creation or measurement of software.

7. I have been a full-time university professor for nearly 30 years conducting peer-reviewed research and teaching at the graduate level. I have held the David M. Roderick Chair of Information Systems at the University of Pittsburgh's Katz Graduate School of Business from 1995 to the present day. I also hold an adjunct professorship at Carnegie Mellon University, where I lecture on software engineering. I was previously a faculty member at the Massachusetts Institute of Technology's Sloan School of Management.

8. I have authored or co-authored more than 70 published articles in scholarly journals and am the editor of two books, one on software project management and one on information systems and industrial competitiveness. This research is consistently highly cited in both the software engineering and the business information systems literature. I have also authored two Harvard Business School-published case studies, including one on the mobile operating systems market.

9. My research has been funded by a variety of organizations, including the National Science Foundation. I have served in senior editorial roles in most of the leading journals in my field, including as Editor-in-Chief of Information Systems Research, Departmental Editor for Information Systems at Management Science, Senior Editor at MIS Quarterly, and Associate Editor at IEEE Transactions on Software Engineering. I have been honored to be named as a Distinguished Fellow of the INFORMS Information Systems Society.

10. Over the last two decades I have been retained as an outside independent expert witness over two dozen times in a variety of computer software-related matters, including software project management, anti-trust, and intellectual property issues.

III. Summary of Opinions

11. Google copied thousands of lines of code and the structure, sequence and organization of the Java APIs without a license to do so. These Java APIs are valuable, copyrighted creative works.

12. The code and the structure, sequence and organization that Google copied is valuable to Google and the Android platform. Google faced market pressure to develop a platform quickly. The copied Java APIs provided pre-written programs that enabled Google to more quickly achieve adoption of Android. The copied Java APIs had been developed, industry-tested, and enhanced over the years, and so reflected a set of high quality resources. A very large community of trained developers had been created and nurtured over multiple years, providing a ready asset available to be deployed on new development. Usage of the copied Java APIs helped Android to become a stable platform in a relatively short amount of time.

13. The copied Java APIs are a critical and central resource within Android. Android will not function without the copied Java APIs, or even without the subset of the specifically copied lines of code, and there is

a significant level of dependency on these copied APIs within Android. In addition, all of the most popular consumer apps that have been developed to run on Android also are heavily dependent on the 37 copied Java APIs.

14. Google viewed the open source licensing options for Java as unacceptable.

15. Google's copying of the Java APIs can be expected to have negatively impacted Oracle's business in both direct and indirect ways. The consistency of the Java APIs has been and continues to be a valuable asset that Oracle, formerly Sun Microsystems, Inc. ("Sun"),[1] protected and promoted through its investments in both licensing and testing. Oracle received no direct licensing revenue or software contributions from Google. Google admits, and independent testing confirms, that Android is incompatible with Java on multiple levels. This incompatibility reduces the size of the Java 'network', including its community of developers who are fragmented relative to what would be the case had Android been a licensed, compatible product.

IV. BACKGROUND ON THE JAVA AND ANDROID PLATFORMS

16. Set forth below is a general summary regarding the Java and Android platforms, and the Java APIs at issue. I have reviewed the more detailed overview of Java and Android prepared by Prof. Doug Schmidt and I rely upon and refer to the background of his expert report. I have also reviewed Mr. Robert Zeidman's report and rely on the code comparison work and analysis in his report.

A. Background on the Java Platform

17. Java is a programming platform first created in 1995 by Sun. In 2010, Sun was acquired by Oracle Corporation, making Oracle the owner and distributor of the Java platform and software products. The Java platform[2] is known to and used by a large and active community of software developers and programmers,[3] who create computer programs or applications,[4] that run on computers, mobile devices and other types of devices for end users (e.g., someone who owns a phone or other type of computing device, and who is a consumer or user of a program or application). Examples of applications include word processors,

[1] The term "Oracle" as used in this report should be read to encompass the activities both of Oracle and Sun, unless specifically stated otherwise.
[2] See Appendix J
[3] The terms (sometimes prefixed by software) "developers," "programmers," "application developers," or "app developers" are used synonymously here to refer to people who write computer programs and develop applications (commonly referred to as "apps").
[4] The terms "programs" and "applications" (or "apps") are used synonymously here to refer to the results of programming activity. In the context of mobile devices and mobile platforms, such as Android or Java on mobile devices, the term "application" is used more frequently. Developers and programmers generally use the word "program" and its synonyms to refer to a structured sequence of organized, syntactic code, end users frequently use the "program" and its synonyms to refer to the applications created by programmers.

spreadsheets, web applications, online games, map applications, search engines or search bars, email tools, calendars, web browsers, weather updates, news feeds, video and media players, and social media applications.

18. Software programmers can create applications for a number of different devices, including desktop computers, cloud computing systems, mobile devices, tablets, e-readers and, increasingly, for automobiles, home appliances, other "smart" devices, embedded devices, and devices in connection with the *Internet of Things ("IoT")*.

19. The Java platform was designed to further the idea of portability, as exemplified by its explicit original motto: "Write Once. Run Anywhere."[5]. In other words, the Java platform enables app developers to write programs for one platform and have them run and behave the same way on any other combination of hardware and operating system that contained the necessary Java platform resources. The platform resources that would be on an end user computer or device are known as the *Java Runtime*. In this way, a programmer would only have to write a program in Java once for a specific Java platform, and it would run across a very broad array of computers, devices and operating systems that contained the Java Runtime for that Java platform.

20. In the Java platform, Oracle provides a set of source code resources called application programming interface ("API") *packages* (and provides the same code in compiled form called "*class libraries*"). The Java API packages are a group of unique, prewritten programs.[6] They contain, for example, organized collections of prewritten classes and methods. These pre-written programs carry out specific, complex coded instructions.

21. Oracle created the Java API packages, and the classes and methods within the API packages, to provide a valuable set of resources to software developers and programmers that makes their work easier and thereby increases the popularity of the Java platform. Java developers and programmers are able to use these API packages, and the classes and methods within them, to carry out the particular functionality they need when they are creating their own programs. If Java developers and programmers elect not to use the API packages, and the classes and methods within them, they would have to write their own classes and methods to carry out the same functionality. But, rather than write their own classes and methods, a developer can type Oracle's proprietary code that takes advantage of the pre-written classes and methods in the API

[5] Fritzinger, J. Steven, and Marianne Mueller. "Java security." White Paper, Sun Microsystems, Inc (1996), available at http://net.uom.gr/Books/Manuals/whitepaper.pdf ("The Java platform also provides an important capability not found in traditional operating systems. This ability, called Write Once/Run Anywhere executables, allows Java programs written on one type of hardware or operating system to run unmodified on almost any other type of computer."); OAGOOGLE0100628653 (2003) ("Write Once, Run Anywhere (WORA) allows developers to write client-side and server-side code and guarantees execution on various platforms.")

[6] The Court of Appeals stated: "Sun wrote a number of ready-to-use Java programs to perform common computer functions and organized those programs into groups it called "packages." United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022, p. 7.

packages. The API packages containing classes and methods are distributed by Oracle in compiled from within the Java Runtime (and are known as the class libraries in that context, once they are compiled). Java programs that have been written with code referencing the API packages in the Java Runtime make calls upon the API packages (class libraries) to invoke the pre-written programs, classes and methods within the API packages.

22. API packages are beneficial to programmers because they allow for the faster, more efficient construction of high quality applications. Rather than engaging in the more laborious task of writing sequences of program code from scratch and determining what can be highly complex permutations of classes and methods themselves, programmers can draw on the resources of the Java API packages within the Java API library (the class libraries) and use the packaged classes and methods inside different Java API packages to more easily create high quality programs.

23. Oracle makes the Java API packages available to developers through a variety of licensing arrangements. The Java API packages are available to developers of applications as part of the Java platform, and as part of the Java platform *Software Development Kit (SDK)* which is sometimes called the *Java Development Kit (JDK)*. Oracle has a licensing regime by which it makes the Java API packages available, for example, to other companies that want to include them in products or other development platforms.[7]

24. As discussed in detail in Prof. Schmidt's report, the Java API packages contain groupings of source code called classes, which in turn contain methods and other elements.[8] Within these source code files there are lines of source code called declaring code, which "declare" information about the method. Oracle created this code to make it easy for programmers to learn sophisticated software design choices allowing the programmers to invoke prewritten programs. Programmers who use the Java platform come to learn this source code that is used to invoke particular methods, and come to know where it is arranged within the overall structure of packages and classes. Declaring code identifies, introduces and specifies packages, classes,

[7] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022, p. 9 ("Although Oracle owns the copyright on Java SE and the API packages, it offers three different licenses to those who want to make use of them. The first is the General Public License, which is free of charge and provides that the licensee can use the packages—both the declaring and implementing code—but must "contribute back" its innovations to the public. This arrangement is referred to as an "open source" license. The second option is the Specification License, which provides that the licensee can use the declaring code and organization of Oracle's API packages but must write its own implementing code. The third option is the Commercial License, which is for businesses that "want to use and customize the full Java code in their commercial products and keep their code secret." Appellant Br. 14. Oracle offers the Commercial License in exchange for royalties. To maintain Java's "write once, run anywhere" motto, the Specification and Commercial Licenses require that the licensees' programs pass certain tests to ensure compatibility with the Java platform.")

[8] See Schmidt Report, Appendix E.

and methods, as well as a variety of other elements.[9] Declaring code is important in realizing the value of the Java API packages because this is how information about the classes and methods within the Java API packages is communicated to programmers and how programmers invoke methods within the API packages.

25. There is also additional code within the classes that carries out the implementation of a particular method, which is referred to as implementing code. The implementing code serves no purpose without the declaring code that describes it and invokes it.

26. Structure, sequence and organization (or SSO) means the taxonomy of the packages, classes, methods and other elements set forth in the 37 Java API packages, including the hierarchy, grouping, complex and intricate relationships and inheritance between and among the foregoing elements and portions of those elements.[10] SSO is important in realizing the value of the Java API packages because it communicates to programmers where to easily find and access the classes and methods that they want to invoke, and how those classes and methods relate to and interact with each other.

27. The Java API packages, and their constituent classes and methods (and other elements) are set forth in the *Java API Specification*.[11] The Java API Specification contains the declaring code for particular methods and other elements, and for example reflects the intricate and hierarchical structure, sequence and organization of the methods within classes and classes within the Java API packages. The Java API Specification is often how developers come to learn the Java API declaring code that is contained in the Java API package source code files and how developers come to learn the organization of the SSO and where to find particular classes and methods. The Java API Specification is a resource that makes the Java API code and Java platform so popular among developers. The Java API packages, both in source code form and in the form of compiled "class libraries" have been worked on since the 1990s and constituted Sun's, and now Oracle's, intellectual property. For example, in 1996, the Java class libraries were described in a 1,650-page book called "The Java™ Class Libraries: An Annotated Reference."[12] The evolution of the API packages in

[9] The Court of Appeals stated that: "Declaring code is the expression that identifies the prewritten function and is sometimes referred to as the "declaration" or "header." As the district court explained, the "main point is that this header line of code introduces the method body and specifies very precisely the inputs, name and other functionality." Id. at 979-80. The expressions used by the programmer from the declaring code command the computer to execute the associated implementing code, which gives the computer the step-by-step instructions for carrying out the declared function." United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022, p. 9.

[10] The Court of Appeals stated that: "Google copied the elaborately organized taxonomy of all the names of methods, classes, interfaces, and packages—the "overall system of organized names—covering 37 packages, with over six hundred classes, with over six thousand methods." Copyrightability Decision, 872 F. Supp. 2d at 999. The parties and district court referred to this taxonomy of expressions as the "structure, sequence, and organization" or "SSO" of the 37 packages. " United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, 1022, p. 11.

[11] *See* Java SE 5 API Specification, https://docs.oracle.com/javase/1.5.0/docs/api/

[12] TX 2488

the class libraries over time is reflected in Appendix G. These libraries continued to provide an easy way for programmers to learn and develop useful and substantial programs. By using the Java API packages in the class libraries, Sun and Oracle contributed to the speed, ease, and quality of application development for developers, and made it unnecessary for programmers to write these methods from scratch. This facilitated deployment of Java applications on a wide variety of devices. (see, e.g., TX 134 at GOOGLE-01-00018143 (illustrating variety of devices on which Java runs)).

B. Background on the Android Platform

28. Android is a combined programming platform and operating system ("Android Platform") offered by Google.[13] It was first announced in 2007, and the first commercial release was in September 2008.[14] I have reviewed documents indicating that the first development of Android occurred at Android, Inc., which was acquired by Google in 2005. Android was originally, and continues to be, used on mobile phones, but is also used in other devices, including wearable devices, automotive devices, and other devices into which Android may be embedded.

29. The Android programming platform contains tools to allow developers to write Android applications, which are designed to appeal to Java developers. Like the Java platform, the Android platform contains a set of API packages. In particular, the Android platform contains the 37 Java APIs (referenced above) that contain the structure, sequence and organization and a great deal of code copied from the Java platform. The Android API also consists of additional API packages.

30. Like in the Java platform, the API packages in Android are a group of prewritten programs that contain prewritten classes and methods, and are included in the platform to facilitate the development of programs and applications by developers and programmers. The API packages in Android are available to developers of applications as part of the Android platform, and as part of the Android platform *Software Development Kit (SDK)*[15]. Applications that developers might develop in the context of the Java platform may alternatively be developed in the Android platform. The purpose of the API packages in Android, including the 37 API packages taken from the Java platform, are to act as pre-written programs that developers and programmers can invoke, rather than having to write their own code from scratch. As detailed below, Google took the Java APIs from the Java platform for use in Android in order to (1) enable quick time to

[13] A representation of the Android platform is at Appendix K. (from GOOGLE-22-00280869-GOOGLE-22-00280977 at GOOGLE-22-00280861. Android is also the name for Google's mobile operating system; A timeline of some relevant events is at Appendix N.

[14] http://techcrunch.com/2007/11/05/breaking-google-announces-android-and-open-handset-alliance/; http://android-developers.blogspot.com/2008/09/announcing-android-10-sdk-release-1.html; Kent German, *A Brief History of Android Phones,* CNET (August 2, 2011), http://www.cnet.com/news/a-brief-history-of-android-phones. <October 2008>

[15] See, for example, http://developer.android.com/sdk/installing/index.html

market, (2) quickly enable high quality development through leveraging the community of Java developers who were already familiar with the Java APIs, and among whom the Java APIs were already popular and (3) enable deployment of applications on a wide variety of devices.

V. THE 37 ORACLE API PACKAGES AT ISSUE WHICH GOOGLE COPIED

31. I have been informed that a copyright owner has the right, under copyright law, to exclude others from reproducing, preparing a derivative work from, distributing, performing, or displaying, the copyrighted work, subject to certain defenses, See 17 U.S.C. §106.

32. I understand that the 37 copied Java APIs represent copyrightable material.[16] As further explained by the Court of Appeals for the Federal Circuit, "the declaring code and the structure, sequence, and organization of the 37 Java API packages at issue are entitled to copyright protection."[17]

33. The 37 Java APIs represent original, creative expressions. As explained by the Court of Appeals for the Federal Circuit:

> "[I]t is undisputed that that the declaring code and the structure and organization of the Java API packages are original. The testimony at trial revealed that designing the Java API packages was a creative process and that the Sun/Oracle developers had a vast range of options for the structure and organization. In its copyrightability decision, the district court specifically found that the API packages are both creative and original, and Google concedes on appeal that the originality requirements are met."[18]

34. Google has, at least, reproduced and distributed thousands of lines of code within the Android API packages, arranged in the same copyrighted structure, sequence and organization as in the Java API packages. It has previously been established that Google copied 37 of them from Java SE5 in all versions of Android through Android version 2.2 ("Froyo" or Android API Level 8).[19] Google has continued this copying in every successive version of Android, from Level 9 through Level 22, as well as in its Brillo and ARC products. In addition to copying the declaring code from the 37 Java API packages, Google has also, in some cases, copied implementing code and private methods not available in the public-facing Java API Specification documentation. It is my understanding that Google's use of this code was not licensed.

[16] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022, pp. 66-69.
[17] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022. p. 69.
[18] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022. p. 21.
[19] See Appendix P

A. Google's Android Version "Froyo"[20] (API Level 8)

35. Java Standard Edition 5 (Java SE 5), released in September of 2004, contained 166 API packages. Of these API packages, it was previously established that Google copied the declaring code and SSO of 37 of them in all Android versions through Android version 2.2 ("Froyo" or Android API Level 8), which "grew to include 168 API packages."[21]

36. The jury found that Google infringed Oracle's copyrights by copying the declaring code and structure, sequence and organization of the 37 Java API packages from Java SE 5 into all Android versions through Android version 2.2 (Froyo/Android API Level 8), and that finding was confirmed by the Court of Appeals for the Federal Circuit. The Court of Appeals found the following:

> "On May 7, 2012, the jury returned a verdict finding that Google infringed Oracle's copyright in the 37 Java API packages." (United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022 p. 12)
>
> "Google conceded that it copied the declaring code used in the 37 packages verbatim." (United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022 page 12.)
>
> "…we agree with Oracle that there was no need for the jury to address copying of the declaring code because Google conceded that it copied it verbatim. Indeed, the district court specifically instructed the jury that 'Google agrees that it uses the same names and declarations" in Android.' (United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022 p. 27)
>
> "…Google copied the declaring source code from the 37 API packages verbatim, inserting that code into parts of its Android software. In doing so, Google copied the elaborately organized taxonomy of all of the names of methods, classes, interfaces and packages – the 'overall system of organized names- covering 37 packages, with over six hundred classes, with over six thousand methods." (United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022. pp. 10-11.)

37. Google has never disputed that it uses the same Java declarations in the 37 API packages in Android. In a prior part of the case, the Court told the jury: "Google agrees that it uses the same names and declarations" in Android. Final Charge to the Jury at 10. Appendix H summarizes the 37 Java API packages that Google was found to have copied into Android (Froyo, API Level 8).[22]

---

[20] A chart summarizing released versions of Android, and corresponding API "level" number and common name (typically releases are named after various desserts), is found at Appendix L.
[21] Docket No. 1202; United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022.
[22] Docket No. 1202; United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022.

B. Google's Android Versions 2.3 ("Gingerbread," API Level 9) through Version 5.1 ("Lollipop," API Level 22)

38. Google has continued to copy the 37 Java API Packages in later versions of Android. In particular, I have overseen code comparison and analysis that demonstrates that both the declaring code and structure, sequence and organization of the 37 Java API Packages is found in Android Version 2.3 (Gingerbread, API Level 9), which was the version of Android that was released immediately after the Froyo version (the subject of the 2012 trial), and the same copyrighted material is found in the recent Android Version 5.1 (Lollipop, API Level 22).

39. I have conferred with Mr. Robert Zeidman on carrying out a code comparison and analysis of these versions of Java and Android and to generate a report measuring the code copying. A summary of the methodology and results that were used in this code comparison is set forth in Mr. Zeidman's report. I also incorporate by reference in its entirety and refer to the contents of Mr. Zeidman's report, reflecting copying of the Java SE 5 code in Android Versions 2.3 (Gingerbread, API level 9) and 5.1 (Lollipop, API level 22). As can be seen in Mr. Zeidman's report, and in particular Exhibits F and S, there are over 11,000 lines of declaring code copied from Java SE 5 into Android Version 2.3 (Gingerbread, API level 9) and 5.1 (Lollipop, API level 22).

40. As can also be seen in Mr. Zeidman's report, the same structure, sequence and organization of the 37 Java API packages are found in Android. As detailed in Mr. Zeidman's report, the copying of the structure, sequence and organization of Java SE 5 code into Android can be seen in several ways:

41. First, from the code comparison charts, comparing Java SE 5 to Android Versions 2.3 (Gingerbread, API level 9) and 5.1 (Lollipop, API level 22), it can be seen that the same method declarations and other declaring code are grouped together in the same classes within both Java and Android, and that those classes are grouped together in the same packages. This shows that the structure, sequence and organization of the 37 Java API packages were reproduced in Android. See Zeidman Report, Exs. F, S.

42. Second, by comparing the API Specification documentation in Java SE 5 and Android (Android version 5.1, Lollipop, API level 22 was used for reference), it can be seen that the same method declarations and other declaring code are grouped together in the same classes within both Java and Android, and that those classes are grouped together in the same packages. This shows that the structure, sequence and organization of the 37 Java API packages were reproduced in Android. See Zeidman Report, Ex. U.

43. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding the code and structure, sequence and organization between Java SE 5 and Android Versions 2.3 (Gingerbread, API level 9) and 5.1 (Lollipop, API level 22) to be accurate. I also worked with a research assistant to

independently reproduce portions of the code and organization analysis above, and obtained the same results. Based on a review of these results, I conclude that the code and the structure, sequence and organization of the 37 Java API packages are reproduced in Android versions 2.3 and 5.1.

C. The Declaring Code and Structure, Sequence and Organization of the 37 Java API Packages in Java SE 5 were also copied in every version of Android between Android Versions 2.3 and 5.1.

44. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding the code and structure, sequence and organization between Java SE 5 and the many Android versions between Versions 2.3 and 5.1 to be accurate. Mr. Zeidman reviewed the "diff reports" showing changes between different versions of the Android API, in order to determine whether any of the code and structure, sequence and organization of that code in the 37 Java API packages was listed on those "diff reports." Based on a review of the results in the Zeidman Report, Exs. Y, Z, AA, BB, CC, I conclude that code and structure, sequence and organization of the 37 Java API packages are reproduced in all Android release versions between 2.3 and 5.1.

D. Android Version 6 (Marshmallow, API Level 23)

45. As can also be seen in Mr. Zeidman's report, the same code and the same structure, sequence and organization of the 37 Java API packages are found in Android Version 6.0 (Marshmallow, API level 23). As detailed in Mr. Zeidman's report, the copying of the code and structure, sequence and organization of Java SE 5 code into Android Version 6.0, can be seen through evidence that the copied lines of code in Android Version 5.1 was also present in Android Version 6.0. (See Zeidman Report, Exs. S, T).

46. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding the code and structure, sequence and organization between Java SE 5 and Android Version 6.0 to be accurate. Mr. Zeidman reviewed the "diff reports" showing changes between Android Version 5.1 and Version 6.0 in order to determine whether any of the code and structure, sequence and organization of that code in the 37 Java API packages was listed on those "diff reports." This process revealed that there were no changes in the copied code or structure, sequence and organization. (See Zeidman Report, Exs. Y, Z, AA, BB, CC). Based on a review of these results, I conclude that code and structure, sequence and organization of the 37 Java API packages are reproduced in Android Version 6.0.

E. Google Continued and Expanded Copying

47. As can also be seen in Mr. Zeidman's report, Google copied code and the structure, sequence and organization from new versions of Java, Java SE 6 and Java SE 7, into Android. As detailed in Mr. Zeidman's report, this can be seen through evidence that declaring code that was introduced in Java SE 6 and Java SE 7

was selectively added to the same classes in Android API levels 9, 11, 14, 19 and 21. (See Zeidman Report, Exs. V, W) Google copied from Java SE 6 and Java SE 7 into versions of Android that were released after the decision by the Federal Circuit Court of Appeals in this case. (See Zeidman Report, Exs. CC, Table 1, 4 and ¶ 121).

48. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding the code and structure, sequence and organization between Java SE 6 and SE 7 and Android to be accurate. Based on a review of these results, I conclude that code and structure, sequence and organization of the 37 Java API packages from Java SE 6 and Java SE 7 are reproduced in Android.

49. The Court of Appeals for the Federal Circuit found that the declaring code and the structure, sequence and organization of the Java API packages are copyrightable. As explained by the Court of Appeals for the Federal Circuit, "the declaring code and the structure, sequence, and organization of the 37 Java API packages at issue are entitled to copyright protection."[23] The declaring code and organization of that code at issue in the Federal Circuit also appeared in the public facing API Specification. I have reviewed, with my research team, the code and organization of code in the classes from Java SE 6 and Java SE 7 discussed here. Like the code at issue in the Federal Circuit opinion, the copied code from Java SE 6 and Java SE 7 discussed here, identifies, introduces and specifies packages, classes, and methods, as well as other elements. For this reason, the code in Java SE 6 and Java SE 7 has the same expressive properties as the code in the classes already addressed by the Court. In other words, the code could have been written in multiple different ways, is creative and serves a similar purpose of declaring, for example, methods. On that basis, I conclude that the code from Java SE 6 and Java SE 7, addressed here, is copyrightable, for the reasons discussed by the Court of Appeals.

50. Similarly, like the code addressed by the Court of Appeals, the additional code from Java SE 6 and Java SE 7 is organized in an intricate hierarchy and elaborately organized taxonomy of expressions. On this basis, I conclude that the structure, sequence and organization of the code from Java SE 6 and Java SE 7 addressed here is copyrightable for the reasons discussed in the Court of Appeals opinion.

F. Google has Copied Implementing Code of the 37 Java API Packages.

51. As can also be seen in Mr. Zeidman's report, some of the code from the 37 Java API packages copied in Android constitutes declaring code from private classes, methods, variables and constructors. See Zeidman Report, ¶¶ 71-75. Private classes are comprised of code that does not occur in the public-facing Java API Specification. Copying of this code from private classes constitutes copying of a type of implementing code. Private classes can be used by developers to implement public API packages. For

[23] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022. p. 69.

example, a developer may be using a publicly declared method declaration in one of the API packages at issue, and the implementation of that public method may – in the course of carrying out what the method is intended to do – rely on and invoke one or more private method declarations, without the developer knowing that this is happening or even being aware that such any private method declaration exists in the code. In this way, the private declarations are part of the implementation of the prewritten programs in the public API packages.

52. The analysis regarding private classes was carried out using the same methodology as was used to compare the public classes in Android Versions 2.3 ("Gingerbread," API Level 9) and Version 5.1 ("Lollipop," API Level 22).

53. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding such code copied from the implementations of the 37 Java API packages to be accurate. Based on a review of these results, I conclude that lines of code from the implementations of the 37 Java API packages are reproduced in Android.

54. The Court of Appeals for the Federal Circuit found that the declaring code and the structure, sequence and organization of the Java API packages are copyrightable. As explained by the Court of Appeals for the Federal Circuit, "the declaring code and the structure, sequence, and organization of the 37 Java API packages at issue are entitled to copyright protection."[24] The declaring code and organization of that code at issue in the Federal Circuit were all code and organization from the source code that also appeared in the Java API Specification. I have reviewed, with my research team, the code and organization of code in the "private" classes discussed here. Like the code at issue in the Federal Circuit opinion, the copied code in the "private" classes discussed here, identifies, introduces and specifies packages, classes, and methods, as well as other. For this reason, the code in the private classes has the same expressive properties as the code in the public classes addressed by the Court. In other words, the code could have been written in multiple different ways, is creative and serves a similar purpose of declaring, for example, methods. On that basis, I conclude that the "private" class code addressed here is copyrightable, for the reasons discussed by the Court of Appeals.

G. Google's "ARC" Product

55. I have reviewed and my technical staff, at my direction, have reviewed source code files and directories regarding Google's App Runtime for Chrome (ARC) product. When ARC is installed on a

[24] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022. p. 69.

Google Chromebook computer, it is able to run Android apps, and thus uses the Android Runtime[25] [26] Detail about the copying of code from the 37 Java API packages into the Android Runtime in Chrome is set forth in the report of Mr. Zeidman. I have conferred with Mr. Zeidman and believe his method of analysis is sound and his results regarding such code are accurate. Therefore, ARC, operating in connection with Chrome and the Android runtime, necessarily reproduces the code and structure, sequence and organization of the 37 Java API packages.

H. Google's "Brillo" Product

56. I have reviewed and my technical staff, at my direction, have reviewed source code files and directories regarding Google's "Brillo" product. As I understand it, Brillo is an embedded operating system, based on Android, and designed to run on "Internet of Things" devices.[27] Within the source code for Brillo there is a directory that has prebuilt (compiled) versions of the Android API packages (i.e. Android libraries), one for each API Level. Within each directory, the files are contained in an archive file called android.jar. As described above, each of the Android API levels copies the declaring code, structure sequence and organization and implementing code from the 37 Java API packages. Therefore, Brillo necessarily reproduces the code and structure, sequence and organization of the 37 Java API packages, as evidenced by the following:

57. I understand that Google, in response to Oracle's written discovery requests, made certain source code available for inspection at the offices of its attorneys. This included source code for Brillo, Google's Internet of Things platform.

58. At my direction, my technical support staff inspected the source code for Brillo on a computer provided by Google's attorneys, to evaluate the extent to which Google copied or reproduced any of the copyrighted works at issue in this lawsuit.

59. Based on this source code review, and my review of documents related to the source code review, it is my opinion that Google copied the code and the structure, sequence and organization from the 37 Java API packages in Brillo. The documentation of this review is provided in Appendix O.

60. Based on this analysis, I conclude that the declaring code and SSO for the 37 Java API Packages from each version of Android is present in the source code for Brillo.

---

[25] http://lifehacker.com/arc-welder-lets-devs-and-you-test-android-apps-in-chr-1694971713; http://www.howtogeek.com/214734/how-to-use-googles-arc-welder-to-run-android-apps-in-chrome/; http://readwrite.com/2015/04/02/arc-welder-android-apps-to-chrome

[26] See, https://developer.chrome.com/apps/getstarted_arc

[27] See, https://developers.google.com/brillo/?hl=en

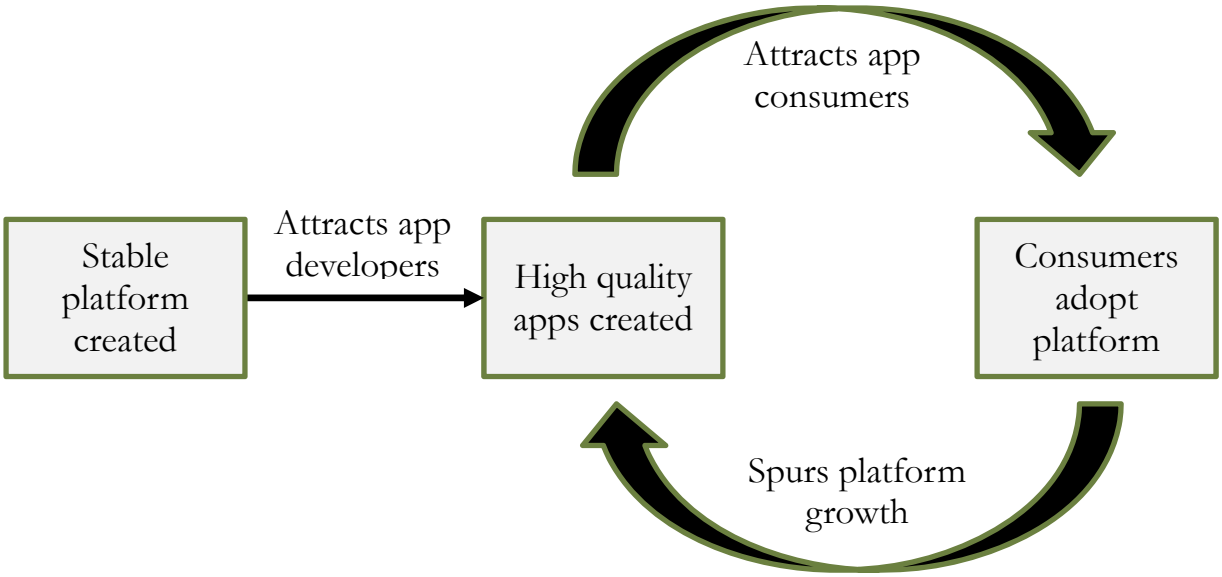
VI. GOOGLE RECOGNIZED THAT THE CODE AND STRUCTURE, SEQUENCE AND ORGANIZATION OF THE 37 JAVA API PACKAGES WOULD BE VALUABLE TO GOOGLE AND TO THE SUCCESS OF ANDROID

61. The Java code and SSO that Google copied are valuable to Google and the Android platform. Android will not operate without the copied Java API declaring code. The Java APIs are central among the APIs in Android, allowed Google to attract developers and more quickly develop Android, and are central to the apps that use the Android operating system. The record demonstrates that other alternatives to unauthorized copying of the Java APIs, such as taking a GPL license, were unacceptable to Google.

62. I have been studying the adoption of software process innovations and the measurement of object-oriented software since the early 1990s, resulting in a number of peer-reviewed research publications. This, combined with my research on economic network effects, gives me a clear perspective on the mechanisms by which software innovations become widely adopted among a community of software developers. In general terms, this can be captured by the diagram in Figure 1 below. An innovative platform is created and introduced to the market. This platform allows for, and mutually benefits, a set of complementary goods. (Common consumer examples include such items as videogame consoles, which attract videogames as their complementary good.) Once high quality apps are available, this makes consumers more likely to choose a platform that runs these apps, e.g. a videogame console with a popular game (the complementary good) motivates consumers to buy that console. Sales of that console then attract other game developers to write for that console, which then further expands the popularity of the console which commences a positive feedback cycle, as shown in Figure 1 below.

63. The relation of this general phenomenon to the facts in this case is as follows. Here, it is the Java programming platform that initially allows the creation of apps. The Java APIs contributed to the Java platform's widespread adoption by developers. The Android programming platform, based on the Java APIs, was used to create the Android mobile operating system (OS). This OS attracts popular apps (the complementary goods), which, in turn, attract consumers who choose a device which runs that mobile OS so that they can use those apps. The sales of those devices then attract other app developers to also write for that platform, which then further expands the popularity of the platform which commences a positive feedback cycle, as shown in Figure 1, below.

Figure 1: The Positive Feedback Loop of API Stability



64. Google benefited from using the copied code and structure, sequence and organization of the 37 Java API packages to leverage their popularity and familiarity among developers, in order to quickly attract developers to the Android platform when it was first created. In addition to this positive effect, by using the Java APIs Google was able to bring a higher quality Android to market more quickly. There is substantial evidence of this motivation, as is summarized by the Court of Appeals for the Federal Circuit: "… Google wanted to capitalize on the fact that software developers were already trained and experienced in using the Java API packages at issue. The district court agreed, finding that, as to the 37 Java API packages, "Google believed Java application programmers would want to find the same 37 sets of functionalities in the new Android system callable by the same names as used in Java." Copyrightability Decision, 872 F. Supp. 2d at 978. Google's interest was in accelerating its development process by "leverag[ing] Java for its existing base of developers."[28]

A. Google's copying of Oracle's 37 copyrighted Java APIs benefited Google

1) Copying the popular and familiar expression of the declaring code significantly increased the speed of developing the Android Platform and bringing it to market

65. Google's copying of Oracle's Java APIs benefited Google by enabling it to more quickly bring Android to market, to attract developers, and to build an app ecosystem, which attracted consumers.

[28] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022 p. 51.

2) Google needed to develop a mobile platform quickly

66. Google needed to develop a mobile platform quickly to establish its presence in the market and to start the process of monetizing data from user engagement with applications and devices.

67. Google is a company with over $350 billion in market capitalization, and a business focused on improving the ways people connect with information. Google provides products and services relating to search, advertising, operating systems, platforms, enterprise and hardware products.[29]

68. A key activity for Google is gathering data "signals" from its users. Users' data signals provide specific information such as geographic location, search query content and web browsing activities, all of which can be collected, organized and sold to third parties, including retailers, manufacturers, service providers, advertisers, market researchers, law enforcement, or any other party interested in understanding how users interact with their digital devices and interfaces.[30]

69. Advertisers, in particular, place high value on the volume, accuracy and frequency of user signal data, as it allows them to plan and execute more effective targeting of advertisements. Advertisements targeted based on user data signals appear in numerous locations including, but not limited to, the advertising columns on search pages, as sponsored previews for audio-visual content, as commercials embedded in streamed content, and as suggestions prompted by the purchasing of applications or other products.[31]

70. Mobile devices such as mobile phones are the main source of a user's mobile data signals. Users generate data signals through the use of web browsers, email applications, navigation aids, messaging applications, video players, streaming music services, games, and a vast array of downloaded applications, both free and purchased. Each such activity generates user data signals that can be collected, organized and sold for profit to third-parties.[32]

71. Usage patterns between mobile and desktop vary quite a bit. In January 2014, mobile devices accounted for 55% of internet activity in the United Sates. Of this, 85% of this internet activity on mobile devices took place via apps; the remaining 15% was through the use of mobile browsers.[33]

---

[29] http://www.forbes.com/companies/google/
[30] Google, https://privacy.google.com/about-ads.html (last viewed January 1, 2016); Christine Erickson, Google Privacy: Five Things the Tech Giant Does with Your Data (March 1, 2012), http://mashable.com/2012/03/01/google-privacy-data-policy/#0nnyMEm_bZqH.
[31] Google, https://privacy.google.com/about-ads.html (last viewed January 1, 2016).
[32] Billy Ehrenberg, *How Much is Your Personal Data Worth,* The Guardian (April 8, 2014), http://www.theguardian.com/news/datablog/2014/apr/22/how-much-is-personal-data-worth.
[33] James O'Toole, *Mobile Apps Overtake PC Internet Usage in US,* CNN Money (February 28, 2014) http://money.cnn.com/2014/02/28/technology/mobile/mobile-apps-internet.

72. Google Services are widely used by a majority of the Android mobile devices. These services include the Chrome browser, Gmail, Google Maps, Google Hangouts, Google+, YouTube, and a virtually limitless supply of products available for purchase and download via Google Play, a retail service for all varieties of mobile applications.[34]

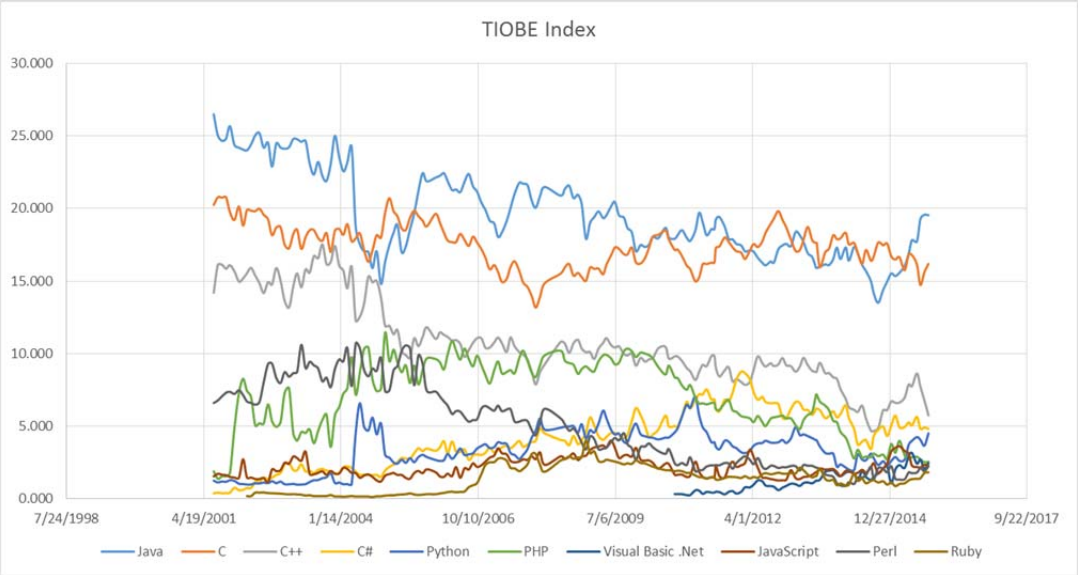
73. Android mobile devices are a significant source of users' mobile data signals. Google's construction of the Android platform is to coordinate with the signal-collection services described above.[35] All else being equal, Google was motivated to get to this market quickly before competing mobile platforms gained significant market share at Google's expense and before it lost the opportunity to dominate the mobile platform so that it could generate revenue from advertising. (Schmidt, Tr. 1456:15-19, 1457:19-25, 1458:1-16.

3) When Android was released, Sun had fostered a large community of developers who were familiar with the expressive content of the Java API Packages

74. Android was able to capitalize on the large, robust community of Java developers and programmers that Sun fostered, and who were already familiar with the expression of the Java APIs. Android continues to benefit from this large community of developers that Oracle has continued to foster since it acquired Sun. With more than 9 million developers, the Java platform takes advantage of the most popular programming language in the world.[36] Moreover, the Java programming language achieved broad popularity well over a decade ago (prior to the regular collection of programming popularity statistics), and has maintained popularity ever since. Figure 2 shows the popularity of programming languages.

[34] Kate Knibbs, *Why Android Phones Now Come With So Many More Google Apps Than Before,* Gizmodo (September 26, 2014), http://gizmodo.com/why-android-phones-now-come-with-so-many-more-google-ap-1639529342.2014).
[35] See e.g.m, GOOG-00273867 (P. 14) and GOOG-00273869 (p. 16).
[36] https://www.java.com/en/about/

Figure 2: Popularity Rankings of Programming Languages, 2001-2015[37]



75. Sun and Oracle worked to refine, further develop, and popularize the Java platform, including the Java API packages, which became an important driver of the overall success of the Java platform. Oracle's history of the Java platform demonstrates how Sun's investment in the Java platform, and the Java APIs that help programmers develop applications, caused the Java platform to become very popular.[38] JDK 1.0 was released in 1996.[39] One year later, in 1997, there were over 220,000 downloads of JDK 1.1 software in just three weeks,[40] and over 2 million downloads by 1998.[41] The 1997 JavaOne event drew 10,000 attendees[42], becoming the world's largest developer conference at that time. Two years later, attendance reached 20,000 attendees.[43]

[37] The TIOBE Programming Community index publishes a monthly indicator of the popularity of programming languages. Ratings are based on the number of skilled engineers, courses, and third party vendors, and are calculated using a variety of search engines. *See* TIOBE Index, http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html; The trends reflected in Figure 2 are corroborated by other widely referenced indexing sites, including redmonk.com and langpo

[38] (http://oracle.com.edgesuite.net/timeline/java/)

[39] *Java SE versions history,* CodeJava, http://www.codejava.net/java-se/java-se-versions-history (Last visited January 5, 2016).

[40] Kieron Murphy, *The pros and cons of JDK1.1,* JavaWorld (April 1, 1997), http://www.javaworld.com/article/2077643/core-java/the-pros-and-cons-of-jdk-1-1.html.

[41] Jason Compton, *Java Time Line,* Chicago Tribune (September 11, 1998), http://articles.chicagotribune.com/1998-09-11/news/9901120092_1_java-developer-conference-javaos-netscape.

[42] *JavaOne Draws 10,000 Attendees—Breaks Attendance Records*, Java Spot News (April 8, 1997), https://web.archive.org/web/19990117032326/http://java.sun.com/pr/1997/april/spotnews/sn970408.html.

[43] Mariva H. Aviram, *What Sun won't tell you about JavaOne,* JavaWorld (August 20, 1999), http://www.javaworld.com/article/2076471/what-sun-won-t-tell-you-about-javaone.html.

4) The Declaring Code and SSO were the key advantage Google leveraged in promoting Android

76. Declaring code and SSO are elements of the Java platform that developers are most likely to be familiar with and appreciate. This is because the ease of learning and the familiarity of their expression helps developers create new applications, making the use of the platform attractive. By contrast, a developer may know less or nothing at all about the implementing code of, for example, a method that is declared by declaring code. The developer will know the declaring code and will have developed expectations about what that expression represents and that the developer can be assured of a particular behavior associated with that expression. Given the direct relationship between declaring code and SSO and developer perception and expectations, a party creating a new platform, attempting to get to market quickly, would specifically benefit from millions of developers' familiarity with this particular material.

77. There is evidence that the value that Google wished to leverage was the familiarity with the expression of the declaring code and SSO. The effect of having the Java API package declaring code and SSO set forth in a specification document (which does not generally contain the implementing code) is to popularize the declaring code and SSO among the developer community, and attract them to the Java development platform. As James Gosling, one of the creators of Java put it in a November 1997 presentation, there are "Two views of an API spec." From the perspective of the "Platform provider" a critical question was "Can I attract developers?" From the "Developer" perspective, a critical question was "What gives me the largest market?"[44] Oracle's executives have also confirmed that the Java API code elements in the context of the specification documentation serve the role of communicating how to use the underlying code.[45]

78. As discussed below in detail, there is significant evidence from Google itself that its adoption of the Java APIs was critical to getting Android to market quickly. There is significant evidence that by using the Java APIs, Google was trying to capitalize upon the large and robust community of developers who were already familiar with the expression of the Java APIs.

79. Early Google documents indicate that Java tools and development platform were familiar to developers. As early as July 26, 2005, a Google presentation on key Android strategic decisions featured a

[44] OAGOOGLE0000021991-OAGOOGLE0000022029 at OAGOOGLE0000022022.
[45] See, Screven, Tr. 512:24-513:2 ("So, look, an API is Java language program text which does two things. It gives form and shape to – to the rest of the Java program, those procedural statements. And it's – informs programmers on how to use those prebuilt Java programs.").

slide on the benefits of Java including "carriers require it…elegant tools story….existing pool of developers and applications."[46]

80. In a January 2, 2006 email, Google engineer Brian Swetland wrote: "[r]easons to shift to a primarily Java API … simplifies the application development story … reduces our development time … faster app development and debuggability. (this is based on experience developing hiptop -- java saved us a pretty crazy amount of time)."[47]

81. Google documents indicate that using Java as the basis for Android would decrease Android's time to market and reduce the risk of a buggy platform. (See, e.g., GOOGLE-01-00075935 at 935 (4/4/2006: "We will ship a more stable product sooner if we do as much as possible in Java").)

82. A July 24, 2006 email from Andy Rubin clearly indicates that the use of Java was necessary to get Android to market quickly: "We were in discussions for 8 months with Sun, walked away, and must prove that our internal effort is clean. Also, because we were in discussion for so long, we must acquire an existing implementation. We ship in 6 months!" TX 147, GOOGLE-01-00023889.

83. In an August 16, 2006 email, Google employees again recognized that using Java was necessary to keep Android on schedule: "if the device is not fast and stable we FAIL"; "we are building a java based system: that decision is final"; "Java not Sun . . . we are building a java based system, not pushing Sun's agendas"; "Any significant change we make will disrupt the schedules." TX 23, GOOGLE-04-00055098 August 16, 2006.

84. A September 28, 2006 Google email observed: "Supporting Java is the best way to harness developers. Fact: Linux fragmentation threatens value. Tools and new app frameworks are biggest hurdles. 6M Java developers worldwide. Tools and documentation exist to support app development without the need to create a large developer services organization. There exist many legacy Java applications. The wireless industry has adopted Java, and the carriers require its support. V Strategy: Leverage Java for its existing base of developers." (p. 10.) TX 158, GOOGLE-01-00025575-587.

85. Additional evidence shows that Google needed to get to market quickly, and the motivation to leverage already mature APIs in the Java platform and the associated community of developers is obvious. (See, e.g., GOOGLE-01-00019529 at 530 ("It is widely believed that if an open platform is not introduced in

[46] GOOGLE-00-00001772, See, e.g., GOOGLE-01-00019511 at 512 (1/2/2006: "Java is more accessable [sic] than C++ . . . . There is more standardization in tools and libraries."); GOOGLE-01-00019527 at 527 (10/11/2005: Android is building a Java OS. We are making Java central to our solution because a) Java, as a programming language, has some advantages because it's the #1 choice for mobile development b) there exists documentation and tools c) carriers require managed code d) Java has a suitable security framework.").)

[47] TX 13, GOOGLE-01-00019511.

the next few years then Microsoft will own the programmable handset platform."); GOOGLE-29-00002338 ("Risk that people may flock to other platforms if we wait too long."); 7/27/2011 Rubin Dep. 179:14, ("I was under incredible schedule pressure . . . ."); 7/27/2011 Rubin Dep. 180:1-12 ("[Y]ou have a window of opportunity in smartphones . . . . You have to ship as soon as feasibly possible. I mean, you go to extraordinary lengths to ship sooner, because it's a very dynamic market. And it could shift directions at any time . . . . So my job as . . . the architect of this business concept was to just do everything that I possibly could to get my solution to the market in the shortest time possible.").)

86. As of September 6, 2006, an email indicates that Google believed that it needed to use the Java API libraries: "is not about you and your work, its [sic] about these guys just feeling now that many of our core libs needed to be in Java." TX 416, GOOGLE-38-00020572

87. Google internal email (Bornstein 2-26-07) says that their aims include providing a "familiar development environment for our third party developers." GOOGLE-38-00015416-17

88. The network effects motivation was cited in a March 1, 2007 Android presentation which indicates that Google was releasing a "Java/Linux Mobile Platform" (p. 43) and planning to "Leverage Java for its existing base of developers." (p. 45.) TX 24, GOOGLE-01-00025375. There are a number of additional documents that are instructive here, as follows:

- Google documents show that there was widespread acceptance of Java among platform vendors. (See, e.g., GOOGLE-12-00003871 at 873 ("Carriers require Java in their terminal.").)
- In one document Google indicated the importance of the network of developers: "Java has very little fragmentation, and it's adoptable. If we play our cards right, we can also leverage not only existing developers, but applications as well."). GOOGLE-02-00111218 at 218.
- Network effects were cited in an August 2007 email from Cedric Beust, a Google engineer specializing in Java, to Rubin: "I can tell you first hand that there are tens of thousands of Java developers who just can't wait to write mobile applications. . . . ." TX 173, GOOGLE-01-00029331 August 16, 2007.
- Similarly, delays in getting Android to market would have posed risk to Android from other competitors, such as Blackberry. (See GOOGLE-00383073 at 075 (discussing BlackBerry users and the likelihood that they would switch to Android as opposed to iPhone the next time they purchased a phone), June 2010). Android needed to offer a platform that could attract developers away from participation in the iPhone market.

89. It is undisputed that attracting the Java developer community was one of the most important reasons why Google chose to use the Java platform tools, in particular the Java API packages, for Android. There is extensive evidence from Google demonstrating that this was the case.

90. Dan Morrill, one of the initial Android developers testified as follows:

> "Q: Let me ask about application developers again for a moment. When Google was developing Android, was it Google's intention to attract Java developers to write -- to write Android applications? … THE WITNESS: By offering a platform that allowed developers to reuse the tools that they were familiar with, yes, that was a goal." (Morrill, Daniel (Vol. 01) - 07/12/2011, 80:4-80:12, July 12, 2011).

91. Dan Bornstein, a key Android developer, stated that they had hired Noser to "implement as much of the standard Java libraries as conceivably makes sense" and emphasized that the declarations were being used to attract developers, giving the following example: "[t]he idea behind this is the same as why were are implementing (say) the standard java.io package: It provides a familiar interface for developers, making it easier for a seasoned Java developer to write original code or port existing code…" GOOGLE-38-00127518. In another example, Dan Bornstein and other Google engineers debated changing the declaring code for a utilities API package, but determined "there's precedent for the name in Java, in the standard package 'java.util' (and subpackages). Us having a 'utils' package would be gratuitously different and would probably cause gnashing of teeth amongst our developer users." GOOGLE-02-00390219. On another occasion, Bornstein stated; "Wherever it's implemented, I think it behooves us to expose a non-jolting familiar-looking Java API for this functionality, which would mean using standard Java classes and interfaces throughout, rather than making a totally separate universe. GOOGLE-02-00384174. This shows that Google chose to use the same declaring code as the Java API packages, because the expression appealed to developers.

92. Bornstein further testified that Android was, in fact, successful in providing familiar API packages to developers: "Q. Using it as you understand it here. A. Well, so I think Android succeeded in providing a familiar enough environment for application developers to use." (Bornstein, Daniel (Vol. 01) - 05/16/2011, 103:22-103:25, May 16, 2011) "Q. And that familiarity is with the Java programming languages and core libraries? A. Some of the familiarity they have would be with the Java programming language. Some of the familiarity they have would be with some of the APIs that are provided as part of the core Android libraries. Some of the familiarity they have would be -- say, the tools that they use, like Eclipse, say." (Bornstein, Daniel (Vol. 01) - 05/16/2011, 104:7-104:15, May 16, 2011) In particular, he testified that the familiarity of the declaring code drove familiarity by developers: "Q. So it made sense to provide a certain familiar set of Java APIs for developers who would create Android applications? A. Well, it made -- it made sense to provide implementations of a set of classes with particular familiar names and methods with particular familiar names along with, you know, to the extent that we could, familiar behavior." (Bornstein, Daniel (Vol. 01) - 05/16/2011, 110:11-110:18, May 16, 2011)

93. More recently, the number three technical executive in Google's Android organization, responsible for the Android runtime and tools, and Google's 30(b)(6) witness, testified that the familiarity of the declarations and organization of the Java API packages drove developer use of the Android platform:

> "Q Do you agree that in 2007 when Android was first released, there wasn't a community of developers that knew about Android initially, correct? A No, there wasn't. Q And so Google had to attract developers to the Android platform through some means in order to get them to write programs for it? A Yeah. Yeah, that's true." (Ghuloum 30(b)(6), Anwar, 90:6-90:14, Dec. 9, 2015)

> "Q In general, in deciding what programming platform to use with respect to Android, do you agree that the number of developers in the world familiar with the platform is a relevant consideration? A Yeah, it's a relevant consideration. There are other considerations as well, but it is -- it is one of the considerations." (Ghuloum 30(b)(6), Anwar, 84:13-84:20, Dec. 9, 2015)

> "Q Is it Google's view that the core – the APIs in the core libraries are known to Java programmers? … THE WITNESS: Yes." (Ghuloum 30(b)(6), Anwar, 16:17-16:21, Dec. 9, 2015)

> "Q -- did -- I mean, is it Google's position that Java programmers who are using the Java platform would recognize, if they have any experience, the -- the -- the core -- the APIs in the core libraries? A Yeah, for the most part. Not all of them but certainly some subset. Q Do you think there's any value to developers in being able to recognize the APIs in the core libraries? A Yes, although it depends on what you mean by "recognize." So if -- if -- I guess, the question is -- actually could you clarify that? What do you mean by "recognize" specifically? Q So by "recognize," I mean the developers using the Java development platform to write a program come to know what the core library APIs are. A Uh-huh. Q And that has value to them as they develop programs? A Yeah. Familiarity, I think, would be a value." (Ghuloum 30(b)(6), Anwar, 17:4-18:2, Dec. 9, 2015)

> "Q Are you saying that familiarity with the core APIs allows programmers to more readily develop programs in the Java platform? A Yeah. … THE WITNESS: Sorry. Probably, yeah, I think that's true." (Ghuloum 30(b)(6), Anwar, 18:19-18:25, Dec. 9, 2015)

> "Q So, for example, the -- as of 2007, the version of the Java programming platform that was available -- you understand that those – that platform contained a set of resources called class libraries, right? A Yes. Q And class libraries are compiled versions of the APIs that we're discussing in this case; isn't that right? A Yes. … Q And when a developer wanted to use the Java programming platform in about 2007, they would download a set of resources that included at least the compiled versions of the Java APIs; is that correct? A Yes. Q And they would -- And you understand -- The 37 APIs that are at issue in this case, you understand those to have existed as of 2007, the ones we're talking about? … Yes, I think so. Q And so what -- When the programmer developing an app using the Java platform in 2007 downloaded the class libraries, including the 37 at issue, would they learn how to use those over time? A I would presume so. If they were using it actively in -- in development and work, yes. Q What kinds of things would the developer have to learn about the class libraries over time in order to become more familiar with them? … THE WITNESS: The -- well, the organization of them into -- into packages, the interfaces and the semantics. … Q So when you said one thing the developers would have to learn over time is the organization of the class libraries in the packages, do you mean to know to look

in one package versus another for a certain, for example, method declaration? A Yeah, that's a good example. Q And so you gave another example, other than organization, that developers would have to learn over time to use the Java APIs. What else did you mention? I've forgotten. A Sorry, the -- the interfaces. Q What do you mean by "the interfaces"? A What parameters you need to call, pass, for example, to the -- to the methods you're using. Q Okay. How does the developer in about 2007, learning how to use the interfaces in the Java API, learn that? … THE WITNESS: So there are a number of ways presumably ranging from documentation to looking at source, looking at examples in other programs. … Q In your experience as one who works with programming platforms, do developers using Java eventually memorize the organization or features of the APIs? A I would imagine that, yeah, that's true. Q Does developers learning of the Java APIs over time constitute an investment by those developers, in your view? A Yeah. I think anything that you learn is an investment. Q And once developers have learned the Java APIs, do you agree that those developers are much more quickly able to develop in the -- in the Java programming platform? . . . THE WITNESS: More quickly. I can't quantify how much more quickly. … Q But, in general, you agree once the developers have come to know the Java APIs, they're -- they're more quickly able to develop applications using that platform? A Yeah." (Ghuloum 30(b)(6), Anwar, 23:3-26:17, Dec. 9, 2015)

"Q Do you believe that developers' familiarities with the Java API is a factor in their comfort with the Java platform? … THE WITNESS: That's a factor." (Ghuloum 30(b)(6), Anwar, 116:16-116:20, Dec. 9, 2015). "Q Do you believe that developers who are using the Java APIs become familiar with, for example, what method declarations -- how they are stated? A Of those Java APIs? Q Correct. A Yes." (Ghuloum 30(b)(6), Anwar, 117:15-117:21, Dec. 9, 2015)"Q What did you have to do in order to familiarize yourself with Android in order to write applications on Android? A Everyone has a different process. I was somewhat familiar with Java already. Of course, I read the documentation and the Android website and downloaded the SDK, which included some samples. NVIDIA had developed their own samples. SDK at the time was using Eclipse-based integrated development, and mostly I just hacked on the samples and changed them and went from there. There was some tutorials as well that I followed. Q Was your familiarity with Java helpful in your beginning to develop for Android? A Yeah." (Ghuloum 30(b)(6), Anwar, 138:20-139:9, Dec. 9, 2015)

94. In summary, and as noted by the Court of Appeals, Google recognized the value in leveraging the large community of Java developers. By using the familiar Java APIs Google both attracted more developers to the Android platform and made the work of those developers easier, thus further accelerating the development and acceptance of the Android platform.

B. Google's copying of the Java APIs enabled it to more rapidly develop a stable API in Android

1) Stable API Packages created a better Android Platform

95. One advantage to Google of copying the code and structure, sequence and organization of the 37 Java API packages would be that those API packages, in addition to having already been written, could be expected to be of higher relative quality because they had been available and in use for a long period of time. Their relative maturity meant they would be relatively more stable than API packages written from scratch.

2) Analysis of API Package stability

96. In order to determine the level to which this is true, my research assistants, under my direction, conducted an analysis of the change behavior of API packages in Android, including those copied from Java. The data required to perform this analysis is a complete specification of the packages, classes, and methods contained in both Java and Android's development kits (the Java JDK and Android SDK, respectively). These two sets of specifications are referred to hereinafter as the "Java APIs" and "Android APIs."

97. First, each version of the JDK between 1.0.2 and 1.8.0 was manually downloaded from publicly available sources.[48] The JDK contains the source files of the various Java API packages at issue, organized into the appropriate hierarchy.

98. Next, Oracle's Javadoc tool was used to convert the organization of the source files into an API documentation file in HTML.[49] The HTML documentation file can then be parsed using a custom-written PHP script and manually verified against archived web pages that provide the corresponding information online.[50] Following these verification steps, the Java API documentation for all versions was assembled into a single tabular form to be used in the actual analysis.[51]

99. For the Android APIs, a file detailing its complete documentation for each version (or API "Level") is available online on Google's Android Git repositories.[52] The documentation for each API Level between 1 ("Base") and 13 ("Honeycomb_MR2") is published as its own XML document. Documentation for Levels 14 ("Ice Cream Sandwich") to 23 (the most current) are published as plain text files. The XML files were converted into tabular form using the Moor XML to CSV tool.[53] The text files were then parsed using a custom PHP script in a manner similar to the JDK.[54] After verifying the results against archived web pages, the results were assembled into their final tabular format.[55]

100. For both the Java APIs and Android APIs, the number of methods changed in each package between two subsequent versions was calculated based on differences between the sets of tabular

[48] Versions 1.1.3 to 1.8.0 were downloaded from Oracle's JDK Archives (http://download.oracle.com/otn/java). Version 1.0.2 was obtained from the Taiwan National Central University JDK Archives (HYPERLINK "http://in.ncu.edu.tw/center5/java/jdk/JDK-1_0_2-win32-x86.exe" http://in.ncu.edu.tw/center5/java/jdk/JDK-1_0_2-win32-x86.exe

[49] See "http://www.oracle.com/technetwork/java/javase/documentation/javadoc-137458.html" http://www.oracle.com/technetwork/java/javase/documentation/javadoc-137458.html

[50] See Appendix F for the full PHP parsing script

[51] See http://www.oracle.com/technetwork/java/javase/documentation/api-jsp-136079.html

[52] See "https://android.googlesource.com/platform/frameworks/base" https://android.googlesource.com/platform/frameworks/base

[53] See "https://xmltocsv.codeplex.com/" https://xmltocsv.codeplex.com

[54] See Appendix I for the full PHP parsing script

[55] See http://www.oracle.com/technetwork/java/javase/documentation/api-jsp-136079.html

documentation data generated in the previous step. For example, the number of changes are calculated for the package "java.util" to indicate the number of method changes that exist in this package from Android API level 1 to 2, 2 to 3, and cumulatively up until Levels 22 and 23. The number of method changes are counted based on the list of methods existing in both continuous API levels, which means newly added or removed methods are not considered as a change in this analysis. For any one particular method existing in both continuous API levels, any change in the method structure[56] is counted as one method change in this analysis.[57] A custom R script was used to calculate the number of method changes for each package between all continuous API levels for both Android APIs and Java SE APIs.[58]

101. The Android SDK is subdivided into three different categories based on their placement in categorical folders—defined by Google—according to function and authorship. These categories are:

102. Libcore – this category represents the critical core of the Android platform and consists of 64 APIs, including 51 APIs developed by Oracle ("Oracle APIs"), 5 APIs developed by Google ("Google API"), and 8 APIs developed by 3rd parties ("3rd Party APIs"). The 51 Oracle APIs in this category include the 37 infringed APIs.

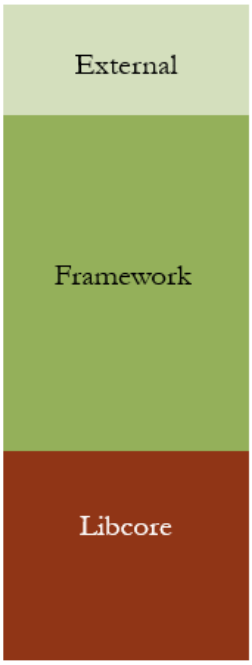
103. Framework – this category consists of 102 APIs, and consists of 100 Google APIs and 2 Oracle APIs.

104. External – this category consists of 34 APIs, including 2 Oracle APIs and 32 3rd Party APIs.

105. These categories are depicted in Figure 3.

[56] Method structure includes method abstract, method synchronization, method exception, method return type, method transient, method volatile, method value, method static, method final, method deprecated, method visibility, and method type. If any of the method structure changes in a method, then the method change is counted as 1.
[57] These include changes in any of the following parameters: method abstract, method synchronization, method exception, method return type, method transient, method volatile, method value, method static, method final, method deprecation or method visibility
[58] See Appendix Q for the R method-counting script.

Figure 3: API Folder Categories of the Android SDK



- **Mostly 3rd Party**
  - 6% Oracle
  - 94% Other
- **Mostly Google**
  - 98% Google
  - 2% Oracle
- **Mostly Java/Oracle**
  - 8% Google
  - 80% Oracle
  - 13% Other

Libcore API package list

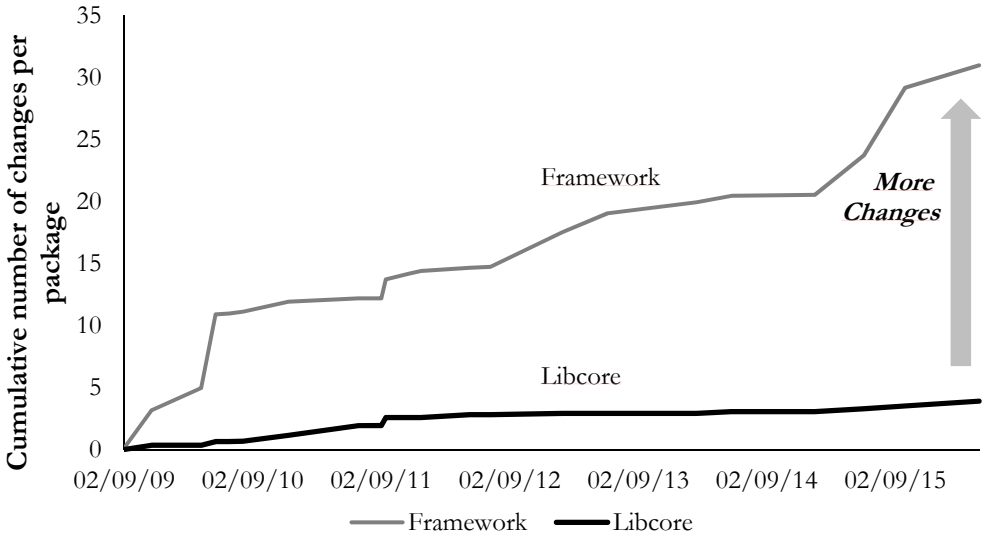
android	java.util.zip
android.system	javax.crypto
dalvik.annotation	javax.crypto.interfaces
dalvik.bytecode	javax.crypto.spec
dalvik.system	javax.net
java.awt.font	javax.net.ssl
java.beans	javax.security.auth
java.io	javax.security.auth.callback
java.lang	javax.security.auth.login
java.lang.annotation	javax.security.auth.x500
java.lang.ref	javax.security.cert
java.lang.reflect	javax.sql
java.math	javax.xml
java.net	javax.xml.datatype
java.nio	javax.xml.namespace
java.nio.channels	javax.xml.parsers
java.nio.channels.spi	javax.xml.transform
java.nio.charset	javax.xml.transform.dom
java.nio.charset.spi	javax.xml.transform.sax
java.security	javax.xml.transform.stream
java.security.acl	javax.xml.validation
java.security.cert	javax.xml.xpath
java.security.interfaces	org.json
java.security.spec	org.w3c.dom
java.sql	org.w3c.dom.ls
java.text	org.xml.sax
java.util	org.xml.sax.ext
java.util.concurrent	org.xml.sax.helpers
java.util.concurrent.atomic	org.xmlpull.v1
java.util.concurrent.locks	org.xmlpull.v1.sax2
java.util.jar	
java.util.logging	
java.util.prefs	
java.util.regex	

106. The Oracle JDK was subdivided into two different categories based on whether they included the 37 infringed APIs or not.

107. Between two subsequent versions the number of method changes in each package for Android SDK and Oracle JDK, respectively, was calculated, and then an average computed for each category. The cumulative number of changes per package in each category was then graphed to visually track the stability of each version over time[59]. In the Android SDK, the Libcore APIs, which are largely composed of Java APIs, stabilized much earlier than the largely Google Framework APIs, and had far fewer cumulative changes per package across all API levels. The Libcore API can be seen to have stabilized after Android API Level 9, which was released in December, 2010. (See Figure 4 below).

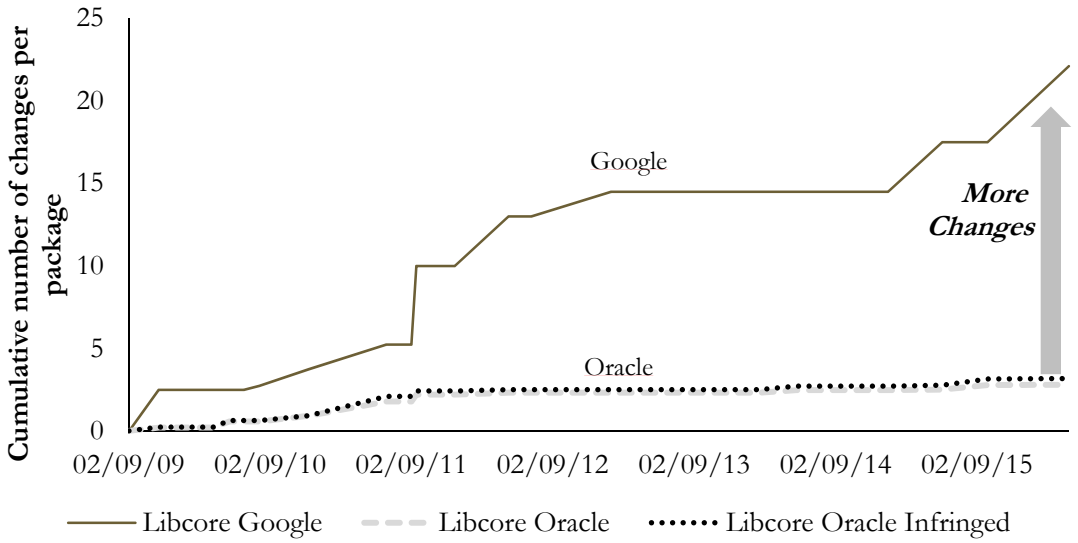
[59] Note that the relatively small number of "External" category APIs are not included in the analysis.

Figure 4: Cumulative Changes per Package (Framework vs Libcore)



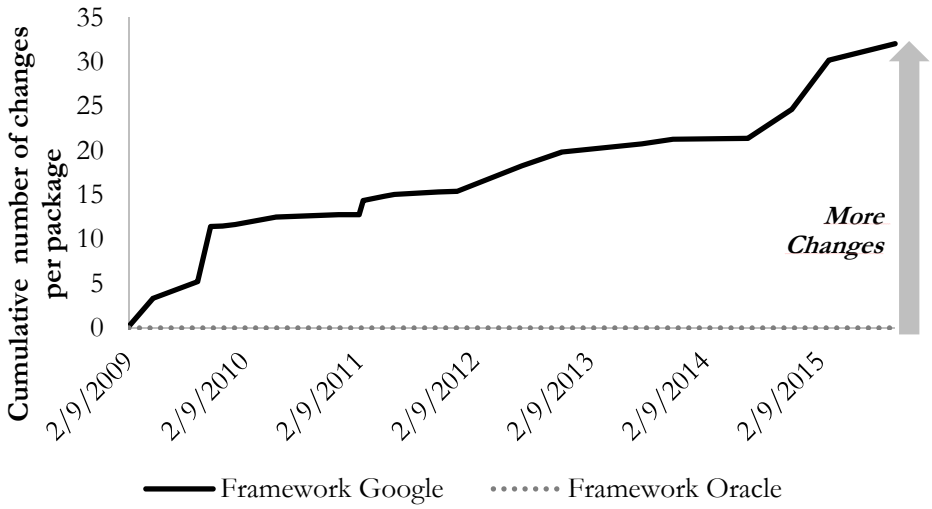
108. The more rapid stabilization of the Android Core compared with its Framework can be attributed to the presence of the Oracle APIs. In the Libcore APIs, the Oracle APIs independently achieved stability much earlier as compared to the Google APIs. In fact, the copied 37 Java APIs (which I will refer to as "the 37 Copied Oracle APIs" or "37 Copied APIs") included in the Oracle APIs represent approximately 73% of the relatively stable Android Core. Moreover, their role in the stabilization of the Android Core is evident when isolating their specifics contributions with respect to stability. As illustrated below in Figure 5, cumulative changes for both the 51 Oracle APIs and only the 37 Infringed Oracle APIs in the Libcore category tapers off beginning with API level 9, which was released in December, 2010, while Libcore Google APIs have many more cumulative changes per package across all API levels and apparently are yet to stabilize, even in the latest version. This suggests that the stability of the Android core is driven by the 37 Copied Oracle APIs.

Figure 5: Cumulative Libcore Package Changes per Package (Google vs. Oracle)



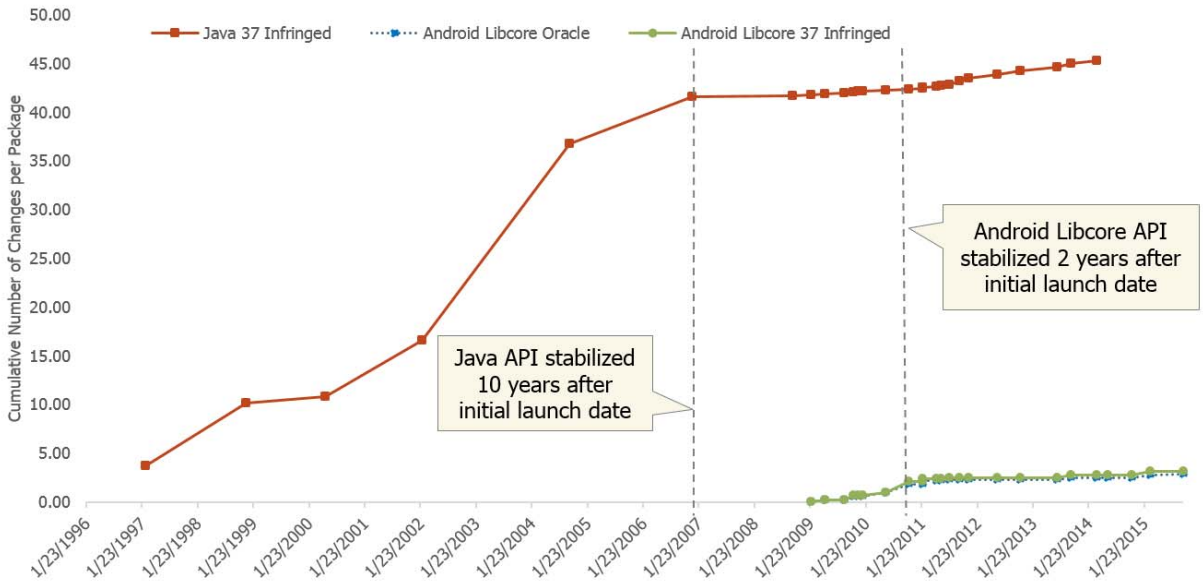
109. The Android Framework, 98% of which is composed of Google APIs, took far longer to stabilize. There are only 2 Oracle APIs in the Android Framework category, and there is no method change for those two Framework Oracle APIs over all API levels. As a result, essentially 100% of the change volatility in the Android Framework APIs arises from Google APIs. As illustrated below in Figure 6, Framework Google APIs have still not demonstrated stabilization, even in the most recent API Level.

Figure 6: Cumulative Framework Changes per Package (Google v Oracle)



110. The same analysis was conducted for all Java APIs in the JDK. As illustrated below in Figure 7, there were a large number of changes in JDK Java APIs before December, 2006. JDK Java APIs can be seen as stabilizing around JDK version 7, which was 10.9 years after the first release of the JDK. In contrast, the Android core APIs, which are driven by the 37 copied Java APIs, stabilized around Android API level 11 in February, 2011, only 2.3 years after the first release of Android. Absent the copied Java APIs it is reasonable to assume that the evolution of the Android core APIs would have exhibited a trajectory more like both the early years of the Java APIs and the early years of the Google-written Framework APIs.

Figure 7: Copying as an 8-Year Market Jump Start[60]

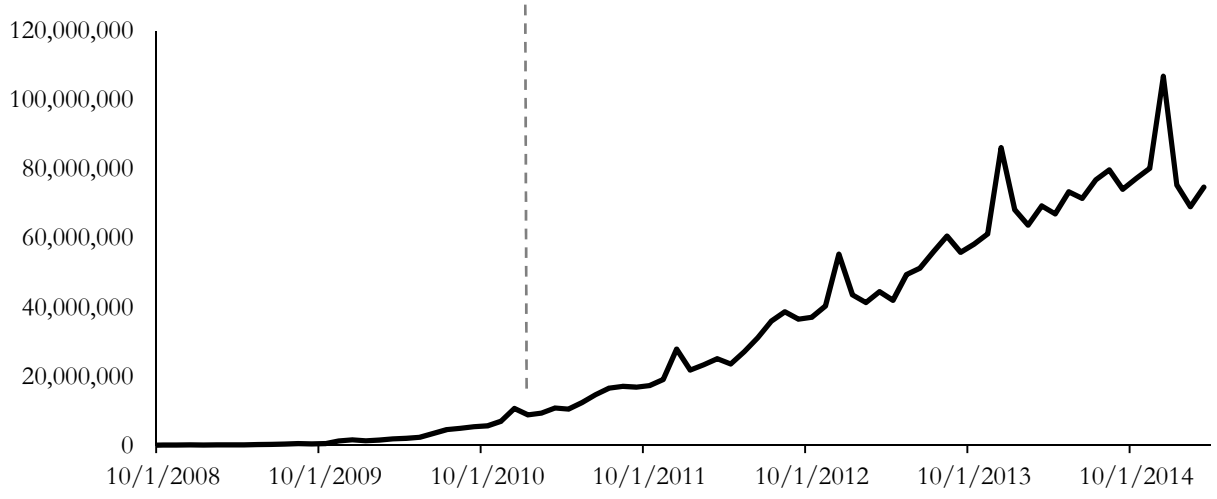


111. The stability of the 37 copied Java API packages contributes to the significant growth of Android. I have observed this growth in three different ways: (1) growth in user activations, (2) growth in the number of active developers and (3) growth of the number of available apps.

112. As I have shown in Figure 8 below, there is a distinct increase in growth in user activations from the release of version API level 9 onwards, which became available in 12/06/2010.

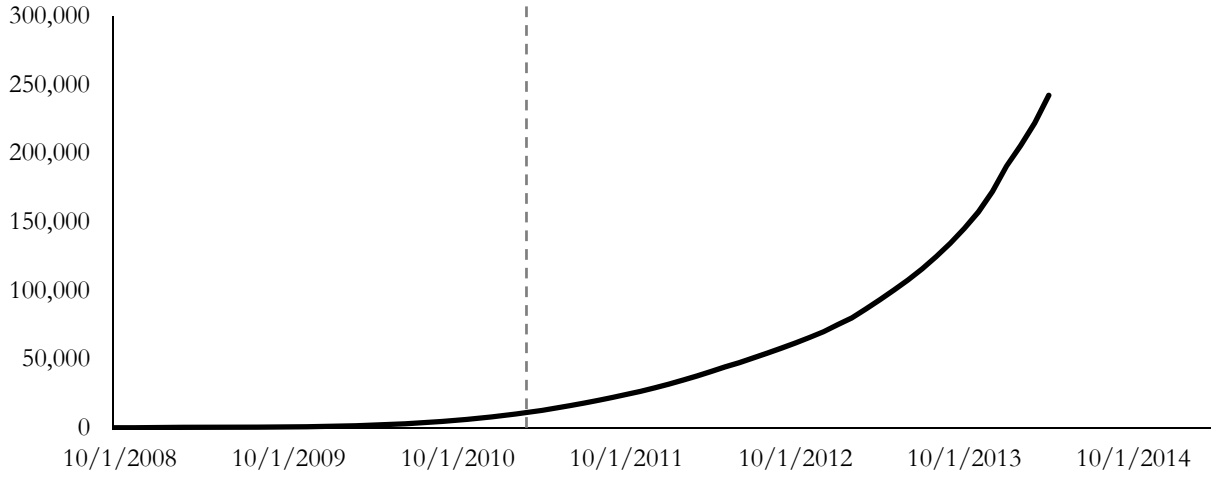
[60] For the curve of the JDK Java APIs, only publicly facing APIs are included since these are the only ones for which the number of changes can be measured. This excludes the following 5 APIs from use in this analysis: javax.crypto, javax.crypto.interfaces, javax.crypto.spec, javax.net, and javax.net.ssl.

Figure 8: Android activations over time[61]



113. As I display in Figure 9 below, here is distinct growth in the number of active developers from the release of version API level 9 onwards, which became available in 12/06/2010.

Figure 9: Android Developer Community growth over time[62]



114. As the two figures above illustrate, Android activations and Android developer community growth, both support the notion that copying of the Java APIs enabled Google to much more quickly release and promote Android than would have otherwise been the case, as well as aiding the acceptance of Android in the

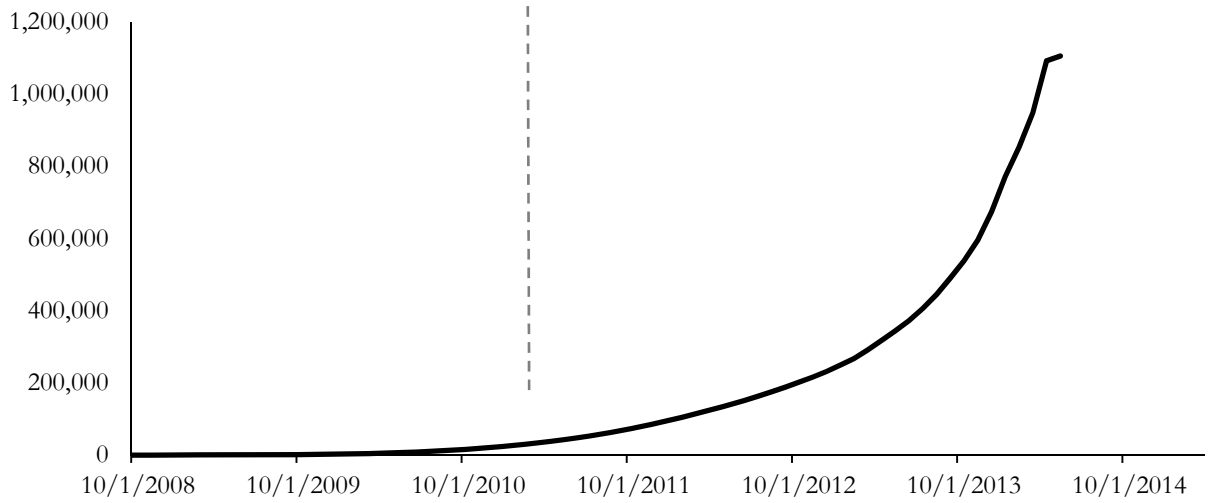
[61] GOOG-00022382
[62] Source: https://github.com/MarcelloLins/GooglePlayAppsCrawler. The date in this analysis is the last release date of the apps in the database

developer community. The graphs suggest that Google may have avoided on the order of eight years of what would have otherwise been relative API instability by copying the 37 time-tested Java APIs.

3) Stable API Packages Helped Fuel the Growth of Apps

115. In a similar fashion, the impact of the stability and availability of Java APIs on the success of Android is also supported by the number of apps available for Android. As I display in Figure 10 below, there is distinct growth in the number of apps available on the market from the release of version API level 9 onwards, which became available in 12/06/2010.

Figure 10: Android Application availability over time[63]



116. Mirroring the positive feedback loop explained earlier in Figure 1, the stability that the 37 Infringed Oracle APIs imparted to the Android platform enabled the development and adoption of apps on the platform, feeding the additional growth of the developer community, which ultimately reinforced the entire cycle. An independent published research paper suggests that the success of applications produced on the Android platform has been linked to the stability and quality of the APIs that are used to create the applications.[64] Low-quality Android apps tend to use Android APIs that have a high change frequency and greater fault proneness. This is because it is difficult for app developers to keep their content working on rapidly changing API versions, which leads to (for example) errors and defects that detract from the end-user experience.

[63] Source: https://github.com/MarcelloLins/GooglePlayAppsCrawler. The date in this analysis is the last release date of the apps in the database

[64] Tian, Y. et al, "What Are the Characteristics of High-Rated Apps? A Case Study on Free Android Applications", *31st IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2015.

117. A senior Android executive testified in deposition that removal of the Java API packages would lead to bugs in applications on the platform: "Q: Do you think that if -- if -- if Google removed the aspects of the Java platform and the Java APIs, in particular, and moved to a new programming environment in which app developers have to adjust to, there would be growing pains?... THE WITNESS: Yes, there would be growing pains. I mean, there's growing pains when we add new APIs and frameworks. … Q Isn't it -- isn't it true that one of the growing pains that Google and -- and Android developers would experience if Google moved away from the Java APIs to a new platform might be bugs in applications? A Yeah, absolutely. We -- we switched our runtime over, and there were growing pains associated with that." (Ghuloum 30(b)(6), Anwar, 150:25-151:17, Dec. 9, 2015)

118. Furthermore, the point at which the Android core API begins to fully stabilize, early 2011, coincides approximately with the point at which the number of Android device activations, active developers, and Android apps begin to accelerate. By August 2010, in a letter to the Google Board of Directors, then CEO Eric Schmidt reflected on early indications of the scale that would later be achieved, noting "For Android, we're now at 200,000 device activations per day, which is a 60% month-over-month growth rate. You start calculating what that will be in a year … and it looks to me as though Android is well past escape velocity at every level." (Google 26-00025769 at 770).

119. Long-term trends in the mobile industry at large clearly demonstrate that an active marketplace of high-quality apps is an important driving factor of that platform's success. Nokia and BlackBerry mobile phones previously held strong positions in the mobile space, with roughly 67% market share in 2009 – around the same time when the app community for the iOS and Android platforms began to grow much faster than those for Nokia and BlackBerry.[65]

120. Since then, Nokia and BlackBerry's market shares have dropped to less than 3% and their presence in the mobile phone space has been all but eliminated. One of the contributing factors to this decline was the relative dearth of BlackBerry apps measured against their competitors. Windows Phone devices' correspondingly weak market position and poor app availability also serve to corroborate this notion. These industry examples stress the important role of the app developer community in bolstering the health of a mobile platform.

121. An Android executive has recently testified that users are attracted to Android because of the number of available applications:

> "Q Isn't it true that one -- one reason that users are attracted to a device platform is the number of applications available on the platform?... THE WITNESS: I believe that is a

[65] Localytics (June 18, 2009), https://www.localytics.com/smartphone-os-wars-develop-for-which-platforms-part-I.

> factor, yes. … Q And so the facility and speed with which a platform like Android can attract application developers, in turn, feeds the number of users?... THE WITNESS: I -- I think it can. I think, though, there are other things that -- well, you're asking me specifically about whether application availability and the speed of application availability on the platform attracts users? … Q Correct. A Yeah, I believe it does have an influence on -- on this. Q And so if suddenly there were -- were less applications available for a platform, that would have some impact on the appeal of the platform, the device platform to users?... THE WITNESS: Yeah. If suddenly, you know, the applications went away or a significant number of applications went away, especially certain, you know, top 1,000 applications went away, that would be problematic. I think that's unlikely to happen, but, yeah, that would be a problem." (Ghuloum 30(b)(6), Anwar, 149:15-150:23, Dec. 9, 2015)

4) API Stability Builds Application Performance, the Developer Ecosystem and End User Engagement

122. As described above, Oracle provided the Java APIs, which permit app developers to invoke prewritten code. Once a Java API has been created, any developer with access to it can make full use of the functionality by invoking associated prewritten code from the APIs.

123. All else being equal, APIs that have been in existence longer are more likely to have been debugged, enhanced, and generally improved until they become relatively mature. As their performance becomes validated and trusted through developer use, the functionality they provide becomes more stable and increasingly useful in producing applications.

124. Newly released APIs have existed for a shorter period, which means they are still subject to regular changes and updates as defects are fixed. New APIs are thus seen as relatively unstable, since the functionality they provide is more likely to be error-prone; it is more difficult to make software function reliably with an API that is constantly changing.

125. Furthermore, since stable APIs produce high-quality apps that attract more users, the growing user base in turn attracts more developers to the platform as well. This catalyzes a positive feedback loop that drastically increases the growth rate of both the user and developer communities.

5) Since Android's Release, the 37 API Packages have been very important to the Android Apps ecosystem and Android developers

126. As described above, platforms, including Android, owe their eventual consumer popularity to the availability of desirable apps. Therefore, an analysis was conducted to assess the relative importance of the 37 Java API packages in the Android app ecosystem from the perspective of app developers. The initial analysis empirically demonstrates the extent to which developers depend on the declarations provided by the infringed packages in the development of Android apps.

a) Methodology

127. The analysis was carried out in four steps: (a) Sampling; (b) Downloading; (c) Reverse-Engineering; and, (d) Analysis.

128. The first step, Sampling, consisted of a selecting a representative sample of important (popular) apps. As of November 2015 there were 128 apps in the Google Play Store with over 100 million recorded downloads (installations), termed "Top Apps".[66] These apps represent the majority of user interactions with the Play Store and have a ubiquitous presence in the Android app ecosystem, and include such apps as Facebook, Messenger, Maps, Gmail, and Google.

129. Following sample selection, the essential package information was downloaded. Each individual app's APK (Android Application Package) was downloaded on 21st December, 2015 from the Google Play Store using the Firefox APK downloader plugin[67].

130. Next, reverse-engineering was conducted by attempting to decompile each APK into its source code in order to extract the underlying dependencies of the app.

131. The contents of each APK file were unpacked, and Android's .dex binary format stored inside was converted into Java's .class binary format[68] using the dex2jar tool. Note that this tool only changes the format of the binary data; the classes and methods within remain incompatible with a Java API. This caused errors in the subsequent steps when the downstream tools tried to read it as a Java source code.

132. Each .jar file was decompiled into the Java source code, (i.e. source files and individual elements from which it was compiled). These decompiled source files ultimately revealed the application's dependencies with respect to Java, Android, or other APIs.[69]

---

[66] The dataset we use in this analysis is created by scraping the mobile app page on the Google Play Store (https://play.google.com/store/apps) with a crawler written in Python. The crawler was run in November 2014. The database covers all Google Play App data from 2008 to 2014. The data available for each app includes several app details including ID, Url, Name, Developer, Publication Date, Category, Free or not, Price, Reviewers, Description, Scores, Last Update Date, Installations and Content Rating.

[67] The Firefox APK downloader Plugin (see https://addons.mozilla.org/en-US/firefox/addon/apk-downloader/) downloads APKs of Google Play Store apps directly to the desktop for use in this analysis. A selenium IDE script and plugin (http://www.seleniumhq.org/projects/ide/) was used to automate the download process.

[68] This step uses the dex2jar tool (see http://sourceforge.net/projects/dex2jar/)

[69] This step uses a tool called CFR – another Java Decompiler (see http://www.benf.org/other/cfr/). This tool accepts the output of the Dex2Jar tool, and de-compiles it into the app's original Java source code. Several Java Decompilers were evaluated. Out of the available Open Source Java Decompilers, CFR was selected due to its ongoing development and support of modern Java 8 lambda features.

133. Finally, the Java source files were processed and analyzed in order to extract the dependency structure of the app's source code.[70] The output of this dependency analysis was subsequently processed to quantify the app's dependency on the 37 copied Java API packages.[71]

134. Due to incompatibilities between Android and the Java tools used to perform the analysis, some apps were unable to be analyzed. First, during the initial download process some apps could not be downloaded due to device and region restrictions. This caused the loss of 5 apps from the 128 app sample. Second, during the dependency mapping process, errors in reformatting and decompilation led to the inability of the Understand tool to map all dependencies. All apps in the report had some percentage of unknown dependencies, as no app was fully compatible with the assumptions of the Java decompiler, but some apps were more severely affected than others. Apps for which over 25% of the dependencies could not be identified were excluded from further analysis. This caused the loss of 17 apps from the sample. Third, during the dependency analysis process the method was not able to programmatically identify which packages belonged to the app and which were from third party sources due to inconsistent naming schemes. This caused 6 apps to be dropped from the sample. Therefore, the final sample consisted of the 100 most popular apps on the Google Play Store that were able to be analyzed. A list of these top apps is presented in Appendix E.

b) Results

135. Based on this analysis of app dependencies it can be demonstrated that the 37 Java APIs are a critical requirement for essentially every one of the 100 top applications. In fact, every one (100%) of the top 100 apps depends upon a minimum of three of the 37 copied Java API packages. The average number of dependencies is 11.5, or nearly a third of the 37 copied API packages. And one of the top 100 apps depends on 23 of the 37 copied API packages.

136. If the analysis is restricted to the most popular of the 100 apps (the 14 apps that are listed as having between 1,000,000,000 and 5,000,000,000 downloads), they can be seen as being even more dependent upon the 37 copied API packages, with the minimum number of dependencies being eight, the average number 13.8, and the maximum number 17.

137. Further, the dependencies for each of the Google-produced apps on the list are depicted in Table 1 below.

[70] The dependency structure was created using the Understand tool (see https://scitools.com/features/#feature-category-dependency-analysis), code analysis software that analyzes calls and references in the app's de-compiled source code to extract its dependencies.

[71] This was performed using a python script on a .csv output. Specifically, the goal was to compute the portion of app-facing dependencies (i.e. the app's architecture depending on another entity) of the 37 Java API packages.

Table 1: Dependencies of Google Apps

Google App	Dependencies
Google Play Books	16
Google Drive	12
Google Play Newsstand	17
Maps	14
Gmail	14
Google Search	15
Google Talkback	8
Google Play Music	17
Google Play Games	12
Google Text-to-speech	15
Google Play Movies & TV	12
Google Streetview	9
Cloud Print	12
Google News & Weather	12
Google Translate	14
Google Calendar	15
Google Keyboard	9
Google Earth	12
Additional Apps	
YouTube	11
Hangouts	12
Chrome Browser - Google	17

138. In order to provide a more complete picture of Google app dependencies, three additional apps are included in Table 1 which are from the list of 28 excluded apps due to data limitations. Their numbers of dependencies are provided for reference only, and are based on less data than the numbers that appear above them.

139. It should be noted that the analysis only looked for dependencies originating from the app developer's code, and ignored any dependencies from code not written by the app developer. As a consequence some dependencies may not be counted, and therefore the method used will tend to be conservative in that it may undercount dependencies.

140. It can be concluded from this analysis that application level dependencies on the 37 copied APIs are significant and central to the Android app ecosystem and its developers as represented by their usage by the most popular apps in the Google Play Store.

C. Centrality of 37 API Packages to Android Source Code

141. Centrality is a metric that is used to describe the relative importance of a particular entity, or node, within a network of interconnected entities.[72] A connection between any two nodes can be described as a dependency. By treating all of the classes in the Android source code as discrete nodes, and all of the connections between classes as dependencies, the entirety of the Android source code can be analyzed as a network, and an impression of the overall contours of influence and interactivity within that network emerges.

142. The centrality of an individual node is roughly proportional to the number of connections that a particular node has to other nodes. An individual node with multiple connections to other nodes will have a higher centrality score than a node with only one connection.

143. Applying the notion of network centrality to the 37 Java API packages within the context of the Android source code as a whole is one way to understand how important those packages are to Android.

144. A high centrality score for the 37 Java API packages would indicate that the classes inside them are connected to a high number of classes in packages outside those packages. Such a pattern would indicate that the rest of the Android source code heavily depends on the 37 Java APIs.

1) PageRank as a means of computing centrality

145. There are a variety of approaches to calculating the centrality of a node, and these measures differ in the balance of inputs that they consider with respect to the qualitative and quantitative nature of nodes and connections. A widely recognized metric is Google's PageRank.[73]

146. PageRank is a centrality measure that was developed by Larry Page and Sergey Brin as part of their research to develop a new search engine, and the mechanics of the PageRank algorithm were published by Page in 1998.[74] It was later implemented in the Google search engine, and is mentioned in the first patents filed describing its search methods.[75]

147. Generally, an entity's PageRank score is proportional to the number of connections that it has, with each connection in turn being weighted by the PageRank of the entity it connects to. This means that calculation of PageRank is a recursive and mathematically cumbersome task. Ultimately, the PageRank scores

[72] See, for instance, http://faculty.ucr.edu/~hanneman/nettext/C10_Centrality.html for a discussion on the notion and implications of centrality using a social network as an analogy.
[73] See http://www.sci.unich.it/~francesc/teaching/network/pagerank for a discussion on the motivation and advantages of PageRank's use in calculating network centrality.
[74] See http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.
[75] See http://www.google.com/patents/US6285999.

of all entities within a particular network form a probability distribution that totals one. For this reason, an entity's PageRank can be thought of as the probability that the random selection of a node inside the network will lead to that specific node.[76]

148. PageRank has since become a widely referenced tool for network analysis, and has been rigorously implemented in fields outside of web search.[77] It has also been used specifically to understand Java networks.[78] For these reasons, it is an appropriate metric for evaluating the centrality of the 37 Java API packages in the Android source code.

2) PageRank analysis and results

149. As mentioned earlier, examining the Android operating system as an interconnected network involves treating every class in the source code as a node in the greater network. Based on this approach, a software tool called Understand is used to analyze intra-class dependencies and discern the broader network structure by building upwards from individual node dependencies.[79]

150. The Android software system network used in this analysis considers every single Java class from Version 5.1.0, release 1 (Lollipop) of the Android Open Source Project (AOSP) source code. The analysis aims to identify the extent to which the Android source code leverages the functionality provided by the 37 Java API packages. The functionality provided by these packages can only be directly implemented by other Java source material. These facts require that only Java material from the AOSP source code be considered in the network analysis.

151. Once created, this network can then be analyzed using a software tool called NetworkX, originally developed by the Los Alamos National Laboratory.[80] NetworkX provides tools for characterizing networks through computational means, including centrality scores.

152. NetworkX was used to analyze the centrality of the Android software system by computing various metrics, including PageRank, for each class inside the network. The results were then processed to identify the scores for only those classes which belong to the 37 Java API packages. The classes in the 37 Java API

[76] See http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf, Section 2.5, for a discussion of the 'Random Surfer Model' that serves as the basis for this interpretation of a PageRank score.
[77] See, for instance, http://people.cis.ksu.edu/~xou/publications/drdc08.pdf (used here to assess vulnerabilities in security networks) and http://www.hindawi.com/journals/tswj/2014/237243/ (used here to understand change propagation in object-oriented software systems).
[78] Puppin, Diego, and Fabrizio Silvestri. "The Social Network of Java Classes." In *Proceedings of the 2006 ACM Symposium on Applied Computing*, pp. 1409-1413. ACM, 2006. http://pomino.isti.cnr.it/~silvestr/wp-content/uploads/2011/02/sac2006.pdf ; OAGOOGLE0000609523; OAGOOGLE0007356223
[79] See https://scitools.com/features/#feature-category-dependency-analysis for a detailed discussion of the code dependency analysis features offered by Understand.
[80] See http://networkx.github.io/.

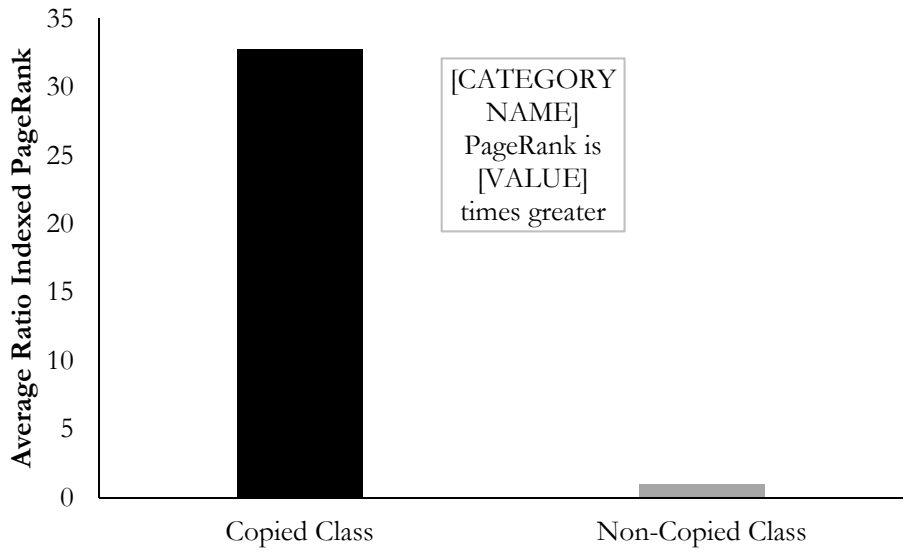
packages could then be compared to the rest of the class groupings across the wider Android source code (i.e. the non-copied classes).

153. Appendix M shows the average PageRank score of both the copied classes and non-copied classes, each of which is normalized by the average score of the non-copied classes. And Figure 11 below very clearly demonstrates the vastly greater PageRank scores of copied classes, with the average copied class boasting a score that is over 30 times greater than that of the average non-copied class. These PageRank results show the importance of the 37 copied Java API packages to the network of the Android operating system.[81]

154. For this reason, the PageRank scores reported in Appendix M are normalized to capture the relative magnitudes of the PageRank scores of copied and non-copied classes. The actual magnitude of the scores themselves is not significant for the reasons stated above.

155. The higher centrality of the 37 Java API packages in aggregate could ostensibly be attributed to the disproportionate impact of a small number of classes in high-ranking results. However, as Figure 11 illustrates, the 37 Java API packages are consistently of high centrality across a deep sampling of PageRank results.

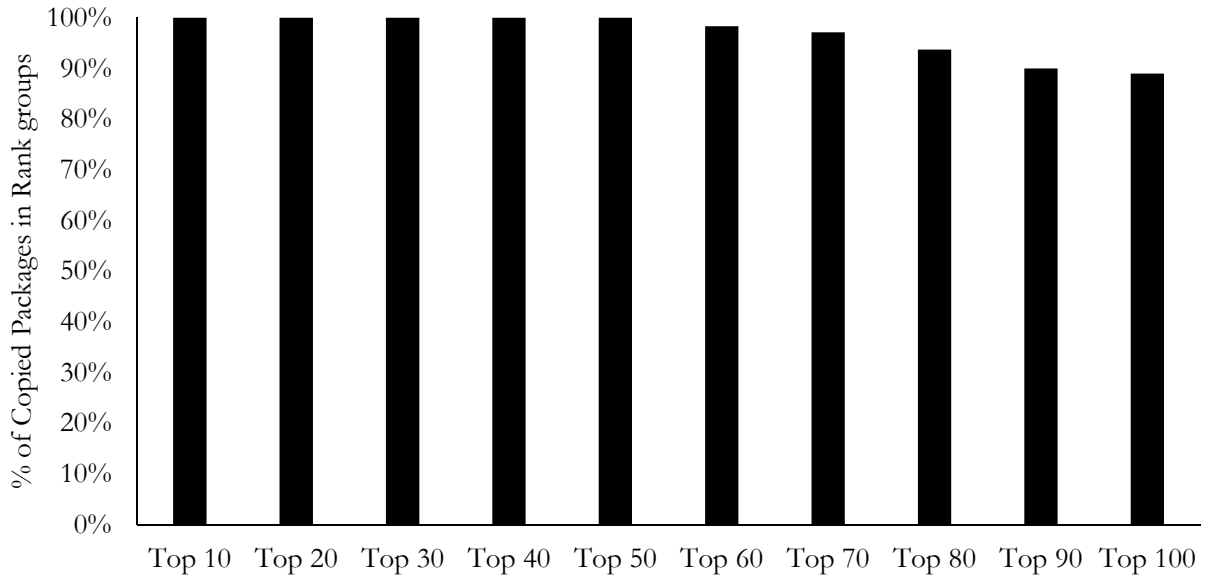
Figure 11: Average PageRank score of a copied class to a non-copied class code



[81] Recall that the PageRank calculation method demands that the sum of the PageRank score of every entity in the network must add up to 1. The Android software system network consists of over 17,000 entities, all of whose PageRank scores must sum to 1. Due to its large size, the PageRank score of any one entity will be very small. This has no implications on the findings of the analysis - it is simply a result of using PageRank on a network of such size.

156. The leftmost column of Figure 12 below shows the percentage of the 10 classes with the highest PageRank score that are copied classes—more importantly, however, the subsequent columns report similarly high percentages. Even when examining the 100 highest-scoring classes, almost 90% of the classes in this list are copied classes.

Figure 12: Percentage of classes with the highest PageRank score that belong to one of the 37 packages for various different rank group sizes



157. Looked at another way, the average page rank score of the copied class is approximately 33 times larger than the average page rank score of other non-copied classes in the Android code[82].

D. The Android Platform will not compile on an Android phone without the declaring code and associated SSO of all or any one of the 37 Java API Packages

158. I have conferred with Prof. Douglas Schmidt who carried out an analysis of, and generated a report about, whether the Android platform properly compiles on a phone without all or any one of the 37 Java API packages. I rely upon and incorporate by reference in its entirety and refer to the contents of Prof. Schmidt's expert report. As detailed in Prof. Schmidt's report, the Android platform won't compile on a phone in the absence of the 37 API packages or any one of them.[83]

159. Google's technical expert witness, Prof. Astrachan, has confirmed that this is the case: "Q. What would happen if you ripped those lines out of Android? A. Well, for the purposes of the Android core

[82] Average for the copied classes was 4.44E-04 and for the non-copied classes was 1.35E-05. PageRank scores for the copied classes appear in Appendix N.
[83] Schmidt Report, ¶¶ 90-97

libraries, those are part of it, so they need to be there for Android to work as it's been designed." (Astrachan Tr. 2212)

E. Google would not accept licensed alternatives

160. As described above, using the Java APIs conveyed significant benefits to Google in terms of the speed of development of Android and its stability. Android and its "top apps" have a high dependency on the 37 Java APIs. Moreover, Sun's open source license was unacceptable to Google.

161. In my professional background, first as a commercial software development project manager and then my academic research and teaching in software engineering management and related consulting/expert witness work, I have a clear understanding of the motivations and risks involved when parties enter into business contracts relating to software. Software development takes a significant amount of skill, and talented software developers tend to be well-compensated. When organizations enter into contracts for software delivery they seek to maximize their risk-adjusted return on this expensive investment. Since the software artifact (the product of the software development process) is an information good, it is what in the business school we term a 'non-rival' good – essentially this means that if I create an information good (a novel, a movie, a piece of software) it has the property that if I sell it to someone (generally in the form of a license to use it), I still possess it. Contrast this with, for example, a typical manufactured good which, once I sell it to someone I have to go create another one – I don't still have the original. This property means that potentially software goods can be highly lucrative, since if I create software that is popular, I can continue licensing it repeatedly without having to recreate it each time. However, the initial software development process is typically complex and costly, and therefore risky, since when starting a project there is no guarantee of a successful completion (technical risk) nor a guarantee of a willing market to buy it (business risk). Therefore, organizations who develop software seek to carefully maximize their returns when licensing their software, including, for example, preserving their rights to continue licensing the software into the future.

162. In my opinion, Sun and Oracle have chosen licenses that provide Sun and Oracle control over the Java API packages and ultimately the platform, and preserved their investment and potential revenue streams. This was achieved either through license terms giving Sun and Oracle control over use of the platform, or terms of non-monetary open source licenses that would require contribution back of code modifications or derivatives. The latter form of license would provide disincentives for some companies that would prefer to be free of such restrictions to take the open source license (and would encourage them to, instead, take a commercial license instead).

163. Google was aware of Sun's open source license for Java, but did not accept that license. A company in Google's position would have incentives that were essentially the opposite of a company in Sun's position.

Google would have incentives to bring a mobile platform to market very quickly and to quickly achieve adoption by many handset manufacturers, and would negatively perceive licensing terms that required relinquishing control, or limiting technical options for Google or its OEM customers, all else being equal. Based on my experience, a platform that was highly controlled by Sun may have created risk for Google in terms of Google's ability to move into the market quickly and to take development of the platform wherever Google wished. Similarly, a company in Google's position would not be amenable to licensing terms (for example, "copyleft" open source licenses) that would create risks for potential OEM customers who might have to relinquish control over changes that they made to the platform. Given that Google was attempting to get initial adoption of the platform, such friction would have been unattractive from a business sense.

164. I have reviewed some of the history of Sun's and Google's interactions regarding an "open source" GPL license, and public documents reflecting Google's position about that license at the time that it was attempting to introduce Android into the market. It is my opinion that Google did not want take such a license because important potential device manufacturer partners likely would not risk accepting code encumbered by this type of license. Again, this behavior is consistent with a party introducing a new platform that needed to be widely adopted by partners. Based on the fact that Google's behavior was consistent with a party attempting to create and quickly bring to market a new platform, it is my opinion that Google would not at the time have found Sun's "open source" license to be a commercially acceptable substitute for the unauthorized copying that Google ultimately carried out. There is historical evidence that this was, in fact, the case.

165. Additional documents show that Google was not willing to accept the "open source" license that Sun offered for Java, which is called the GPLv.2 license with Classpath exception (here simply the "GPL" license). During negotiations in February 2006, Sun offered this GPL license to Google, but Google did not accept it. OAGOOGLE0001817230.[84]

166. For example, in an August 11, 2007 email, Andy Rubin, Google's head of Android, explained to key Android engineers why Google could not use Sun's version of Java licensed under the GPL license: "...and as far as GPL-ing the VM, everything that is linked with the VM would get infected. The problem with GPL in embedded systems is that it's viral, and there is no way (for example) OEMs or Carriers to differentiate by adding proprietary works. We are building a platform where the entire purpose is to let people differentiate on top of it. Finally, Sun has a different license for its library for SE and ME. The SE library is LGPL, ME library is GPL. That means anything that links with the ME library gets infected. And the SE library is not optimized for embedded systems. Sun chose GPL for this exact reason so that companies would need to

[84] Sun and Oracle eventually offered the GPL licensed code as part of a project called "OpenJDK," in May 2007. http://mail.openjdk.java.net/pipermail/announce/2007-May.txt

come back to them and take a direct license and pay royalties. Tricky, no?" TX 230, GOOGLE-02-00020474.

167. On January 4, 2007, Dan Bornstein, one of the primary Android engineers, expressed the view that the GPL license "makes it impossible to create 'extended subsets' of the libraries based on the GPLed version" of Java. GOOGLE-02-00079838 (Ghuloum Depo. Ex. 5018). Google's witness regarding the viability of GPL code admitted that as of 2007, Google was concerned about applying GPL to Android: "Q So it's correct that this January 4, 2007, e-mail from Dan Bornstein to others at Google is discussing the GPL license, correct?... THE WITNESS: It appears to be discussing GPL license. Q And near the bottom of the e-mail about two-thirds of the way down, Mr. Bornstein writes: "This makes it possible -- impossible to create 'extended subsets' of the libraries based on GPL'd version, which is something that Sun has historically feared." Do you see that? A Yes. Q So is it your understanding on behalf of Google that this e-mail is discussing making it impossible to create extended subsets of the Java libraries based on the GPL version of Java?... THE WITNESS: They appear to be discussing that. … Q And so don't you agree that it's correct that as of 2007, Mr. Bornstein was expressing concern about the GPL as applied to the Java libraries?... THE WITNESS: Yes, that does appear to be the case." (Ghuloum 30(b)(6), Anwar, 130:12-131:19, Dec. 9, 2015)

168. In my opinion, Google would not take the GPL licensed version of Java as the basis for Android because it could be expected (and create risk) that a GPL licensed version of Java would have required Google and Android phone OEMs (such as Samsung, HTC and LG) to contribute back to the open source community (and offer under a GPL license) modifications and additions that they made to the API packages, as well as other derivative works that they created containing portions of the GPL licensed code (or at least there is evidence that there was sufficient concern with respect to these requirements).[85]

> In this way, modifications or derivatives of the APIs in the core libraries created by phone OEMs would, themselves, be subject to the terms of the GPL license and could not be withheld by the phone OEMs as proprietary - product differentiation being part of OEM competition. If Google had taken the GPL licensed version of Java, but Google or its phone manufacturer customers had tried to hold back any modifications that they made to Android, I have been informed that they would face the risk of a copyright lawsuit, since that activity could cause the license to terminate.[86]

[85] For example, if Google or an OEM made additions to the core library APIs, those would have to be licensed under GPL. (Ghuloum 30(b)(6), Anwar, 32:1-33:25, Dec. 9, 2015). Similarly, OEMs make may changes to the application frameworks or native libraries portions of the Java platform. Further, the Android platform stack has sets of APIs called the *application frameworks* and the *libraries*, in addition to the core libraries containing the 37 Java API packages. (These can be seen in the representation of the Android platform at Appendix L) Phone manufacturers, such as Samsung or HTC, may want to make additions or changes to APIs in the application framework or libraries in Android, which extend, implement or otherwise modify or combine with the APIs in the core libraries. (Ghuloum 30(b)(6), Anwar, 41:8-43:16, 125:16-129:14, Dec. 9, 2015) (for example, testifying that APIs in the applications frameworks may create subclasses of classes in the core libraries or implement interfaces in the core libraries)

[86] Discussion of lawsuits brought against software companies for failing to comply with the GPL license can be found in the following: https://opensource.com/law/14/7/lawsuit-threatens-break-new-ground-gpl-and-software-licensing-

Thus, Google refused to accept the GPLv.2 license that was offered by Sun. [redacted][87]

169. Google explicitly and publicly stated its reason for avoiding the GPL license when Android was released. For example, on Google's Android licensing page, Google points out that it chose to apply the more permissive Apache license to Android, instead of the LGPL license (a variant of the GPL license that I understand to be similar to GPLv.2 with Classpath Exception) because the LGPL license would have caused business risk and uncertainty for OEM handset manufacturers and other business partners with respect to restrictions on their ability to withhold their software designs as proprietary. Google observes that handset manufacturers would be trying to avoid having the GPL terms apply to their own changes or designs, because those companies would not want to risk making their own code subject to the "viral" – as Google Android executives term it[88] – effect of the GPL license. In particular, the page states the following:

170. "LGPL (in simplified terms) requires either: shipping of source to the application; a written offer for source; or linking the LGPL-ed library dynamically and allowing users to manually upgrade or replace the library. Since Android software is typically shipped in the form of a static system image, complying with these requirements ends up restricting OEMs' designs. (For instance, it's difficult for a user to replace a library on read-only flash storage.)"

171. "LGPL requires allowance of customer modification and reverse engineering for debugging those modifications. Most device makers do not want to have to be bound by these terms. So to minimize the burden on these companies, we minimize usage of LGPL software in userspace."

172. "Historically, LGPL libraries have been the source of a large number of compliance problems for downstream device makers and application developers. Educating engineers on these issues is difficult and slow-going, unfortunately. It's critical to Android's success that it be as easy as possible for device makers to

---

issues; http://www.pcworld.com/article/2893852/vmware-sued-for-alleged-gpl-license-infractions.html; http://torquemag.io/busybox/

[87] [redacted] (Gupta, Vineet (Vol. 01) - 07/26/2011, 189:3-189:20, July 26, 2011)

[88] TX 230, GOOGLE-02-00020474.

comply with the licenses. Given the difficulties with complying with LGPL in the past, it is most prudent to simply not use LGPL libraries if we can avoid it."[89]

173. To understand the implications of these paragraphs, it is important to consider the concept of "*userspace*." When Google refers to userspace it is speaking of all layers of code of a platform that are above the operating system "kernel."[90] In particular, in the case of Android, the kernel means the "Linux" kernel, and "userspace" means the code of all of the various APIs and application frameworks in the Android platform stack above the Linux kernel. (These can be seen in the representation of the Android platform at Appendix K. Google itself defines userspace as "non-kernel" software. As Google puts it: "We are sometimes asked why Apache Software License 2.0 is the preferred license for Android. For userspace (that is, non-kernel) software, we do in fact prefer ASL2.0 (and similar licenses like BSD, MIT, etc.) over other licenses such as LGPL." https://source.android.com/source/licenses.html This suggests that Google wanted to protect all of the code in userspace from having to be GPL licensed. For example, a 2008 Google presentation discusses licensing issues regarding Android and acutely asserts: "we want to keep GPL out of user-space."[91]

174. Industry observers have explained that if the GPL license were applied to Android, it would cause difficulties because it would require OEM handset manufacturers to also apply the GPL license to their own modifications to Android code, thus preventing those companies from holding their own code back as proprietary. At the very least, applying GPL to Android would create significant uncertainty or risk for handset manufacturers in this regard. For example, a 2007 Ars Technica article by Ryan Paul, shortly after Android was released, observed that a major driver of Google's choice not to use the GPLv2 license was to protect the ability of handset makers and carriers to hold their modifications to Android as proprietary:[92]

175. The Apache license, chosen by Google "is widely used in the open-source software community and has been approved by the Open Source Initiative, is a permissive license that is conducive to commercial development and proprietary redistribution. Code that is distributed under the ASL and other permissive licenses can be integrated into closed-source proprietary products and redistributed under a broad variety of other terms. Unlike permissive open-source licenses, "copyleft" licenses (such as the GPL) generally impose restrictions on redistribution of code in order to ensure that modifications and derivatives are kept open and distributed under similar terms."[93]

---

[89] http://source.android.com/source/licenses.html (visited 12/29/2015).
[90] https://en.wikipedia.org/wiki/User_space.
[91] GOOGLE-22-00280869-GOOGLE-22-00280977 at GOOGLE-22-00280894.
[92] http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/
[93] http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/

176. "Permissive licenses like the ASL and BSD license are preferred by many companies because such licenses make it possible to use open-source software code without having to turn proprietary enhancements back over to the open source software community. These licenses encourage commercial adoption of open-source software because they make it possible for companies to profit from investing in enhancements made to existing open-source software solutions. That potential for proprietary investment on top of an open stack is most likely what inspired Google to adopt the Apache Software License for its mobile platform. Availability of Android under the ASL will ensure that a broader number of companies will be able to adopt the platform and build on top of it without having to expose the inner workings of proprietary technologies that give them a competitive advantage."[94]

177. "Although using a permissive license like ASL is the best way to build support for the Android platform, critics argue that Google has sacrificed an opportunity to encourage greater openness in the broader mobile software space. If Android was distributed under the GPLv2, companies building on top of the platform would have to share their enhancements, which could theoretically lead to widespread sharing of code and a more rapid acceleration of mobile software development."[95]

178. "The counterargument is that distributing Android under a copyleft license could potentially limit the evolution of the mobile software ecosystem by discouraging commercial development on top of the platform. Proprietary mobile software development companies that integrate Android into their technologies would have to dramatically change their business models if they aren't given the ability to keep their enhancements proprietary."[96]

179. "When it comes right down to it, the handset makers are the developers who are most significantly affected by the Android license, since they are the primary distributors of mobile phone platforms. The ASL will allow individual handset makers to develop proprietary customizations for the platform as needed to accommodate the unique technologies in their individual products."[97]

180. Google explicitly endorsed this explanation of why it would not accept the GPL license as applied to Android. On the Android website, Google stated: "Why are you releasing the code under the Apache License instead of GPLv2? One of the best explanations for the reasoning behind releasing code under Apache2 can be found in a ArsTechnica article by Ryan Paul," and linked to the article discussed above.[98] Internal discussion at Google shows that this article was approved by Andy Rubin, the head of Android. A

[94] http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/
[95] http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/
[96] http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/
[97] http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/
[98] https://web.archive.org/web/20090124043917/http://code.google.com/android/kb/licensingandoss.html#apache2

November 11, 2007 IM exchange between Google employee David McLaughlin and Chris DiBona, the head of open source at Google, proposed the language endorsing the article and states that it was "suggested by andy yesterday" and "that was a great article." GOOGLE-14-00025664.

181. Another 2012 published article indicates that industry analysts did not believe that Google could release Android under the GPL license because "someone is going to have to write or port a lot of low-level code to access mobile specific capabilities related to smartphones. GPL licenses often restrict vendors from monetizing the code in certain ways without making it also open source, so that might be another factor."[99]

182. Based on the concerns expressed above, I believe that handset manufacturers in 2007-2010 would have resisted use and distribution of GPL licensed versions of Android, as it would limit the handset manufacturers' ability to withhold as proprietary their own modifications to Android and would therefore create a business risk for them. For example, a Google executive testified that Google would have to get OEMs to agree in advance that they were willing to subject any new APIs they created to GPL, but could not say with certainty that OEMs would agree to that. (Ghuloum 30(b)(6), Anwar, 35:14-36:3, 58:4-58:8, 58:17-59:10, 62:14-62:19 Dec. 9, 2015). The risks of OEM concerns would have been particularly acute at a time of initial introduction of the platform when its success was uncertain. As of December 2015 Google had not used GPL-licensed code in any prior version of Android. (Ghuloum 30(b)(6), Anwar, 27:24-28:5, 30:9-30:14, Dec. 9, 2015).

VII. ANDROID IS INCOMPATIBLE WITH JAVA AND GOOGLE'S USE OF THE 37 API PACKAGES FRAGMENTED JAVA

183. Based on my understanding of the importance of stable, compatible and widely adopted technology and the mechanisms by which such technology is created, maintained and licensed it is clear that part of the market success of Java has been Oracle's control of the Java platform and its rigorous compatibility enforcement through mechanisms such as licensing and testing. Android's unlicensed use of portions of Java fails these tests making Android incompatible with the Java platform. This makes software written for one, generally incompatible with the other, fragmenting the Java standard. This fragmentation disrupts the Java developer ecosystem and reduces the effective size of the Java developer network, reducing its value.

184. Compatibility within defined Java specifications has been an important element of value to Oracle from the Java platform. In general, compatibility enables more developers and more users to use the Java platform in a variety of contexts, thereby increasing the network that is using Java, which in turn increases the value of the platform and provides Oracle more opportunities to monetize the platform—i.e. network effects.

[99] http://www.javaworld.com/article/2078666/mobile-java/open-source-java-for-android--don-t-bet-on-it.html

For example, as discussed, Java enables applications to run in a variety of computing devices, independent of the idiosyncrasies of a given device's processor architecture.

185. Compatibility affords developers relative certainty that Java applications they develop will, in fact, consistently run across a variety of devices and that they will have access to a large community of users associated with such devices. Compatibility also reduces the costs of software development for such developers, in that they do not have to substantially invest in redeveloping software for a variety of underlying hardware or operating system contexts.

186. Compatibility also affords users relative certainty that Java applications that they acquire will run across a variety of devices. Thus Java compatibility enables availability of a greater number and variety of applications to users.

187. Java compatibility creates incentives for more developers to invest in writing programs for the Java platform and incentives for users to acquire and use such applications.

188. Oracle has taken steps to enforce compatibility within the Java ecosystem, and to maintain the "write once, run anywhere," proposition of Java, and to obtain the benefits discussed above, including those listed below.

- Sun and Oracle defined API specifications for particular versions of Java (for example, Java SE 5, SE 6 and SE 7), which define the set of API packages that are to be implemented and how they are to behave.

- Sun and Oracle maintain the Technology Compatibility Kit (TCK), which is a suite of tests against which licensed implementations of the Java API packages are to be tested for compatibility.

- Sun's and Oracle's agreements require that implementations of the Java platform conform to the set of APIs set forth in the specifications and prohibit implementations of the Java platform that include a subset or a superset of the API packages. For example, the "Specification License" that applies to the Java SE 5 API specification requires that any implementation of the Java SE 5 API specification: "(i) fully implements the Spec(s) including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the Licensor Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space other than those required/authorized by the Specification or Specifications being implemented; and (iii) passes the TCK (including satisfying the requirements of the applicable TCK Users Guide) for such Specification. The foregoing license is expressly conditioned on your not acting outside its scope. No license is granted hereunder for any other purpose." TX 610.1. Sun and Oracle's commercial licenses also had these requirements. TX 498 ("may not subset or superset the Java Classes" and requiring "compatibility testing" against the "applicable TCK")

189. As discussed below, there is significant evidence that through its use of portions of the Java API packages in Android, Google has introduced an incompatible "Java-like" platform. As discussed, developers are likely to have familiarity with the expression of the 37 Java APIs that were copied, and this will draw them to use of the Android platform and enable them to start their work quickly. While Google benefits from the use of the Java APIs in this way, Oracle's benefits from network effects inherent in Java compatibility are reduced, as the "write once, run anywhere" proposition and all the benefits of compatibility discussed above are impaired. It is ultimately the case that programs developed within the context of Android (based on developer familiarity with the expression of the Java APIs), do not run on the Java platform. Thus, applications that Oracle could have otherwise expected to benefit from, because they would add to the appeal of Java to end users, drive increased scale of the platform by users, and further increase investment by new developers, are outside of the Java network.

190. Further, from a developer perspective, as developers become familiar with the "Android version" of the copied Java packages, there is risk that they will begin to see the full and compatible set of API package resources in Java as resources that are less necessary to remember and leverage, because their knowledge of the 37 Java APIs reproduced in Android was sufficient for them to get started working in Android and is all they need to continue. There is risk that this dynamic will require further investment by Oracle to keep the full and compatible set of API packages known to developers who were aware of Java before they began working in Android. Further, there is risk that new developers in Android, who have their first exposure to Java in the context of Android, will believe that the incomplete and incompatible set of Java API packages copied in Android, which are placed among a number of other different API packages, are all that is necessary to know about developing for the Java platform. Oracle would have to make investments (that they would not otherwise have to make) to specifically educate new developers regarding the differences between the Java API packages in the Java platform and the incompatible packages in the Android platform. This, too, risks causing friction and increased cost to Oracle in maintaining the Java network, where such risk and forced investment would not exist if, rather than copy an incompatible subset of the Java API packages, Google had created Android as licensed and fully Java compatible

191. As discussed below, there is considerable evidence that Google has introduced incompatibility between Android and Java, both in terms of developer expectations and in terms of actual operation of the API packages, which has led to the foregoing risk and cost to the Java network.

A. Android Creates An Incompatible "Subset" and "Superset" Of the Java API Packages.

192. There does not appear to be any dispute that, in Android, Google both used a "subset" and a "superset" of the Java API packages.

193. First, Google used a "subset" of the Java API packages, in that it used 37 of the API packages from the 166 total API packages in Java SE 5. A chart depicting the total API packages in Java SE 5, with highlighting reflecting the subset copied by Google is attached at Appendix H.[100] In exchanges between Noser and Google engineers who worked on Android, they referred to the Android API as a "subset" of "JDK 1.5" (i.e. Java SE 5). GOOGLE-02-00071778-79. Documents show Google engineers choosing which Java packages they would use in Android. GOOGLE-01-00023872, GOOGLE-01-00025523

194. Second, Google used a "superset" of the Java API packages, in that it added packages: javax.microedition.khronos.egl and javax.microedition.khronos.opengles, which are not part of the "javax" namespace in Java SE.[101] Similarly, as discussed in Prof. Schmidt's report, there are number of instances in Android where elements such as extra methods are added to existing classes in the Java SE namespace. This too constitutes an impermissible "superset" of the Java API packages.

B. Google Recognizes that Android is Incompatible with Java and that Android's Use of Java Reduces Interoperability and Compatibility

195. There is substantial evidence from Google that Android is not compatible with Java and that Android's use of Java reduces interoperability and compatibility within platforms that permit programmers to write programs for the Java platform.

196. Dan Morrill, one of the earliest Android engineers, testified that Android is not compatible with Java: "Q. I want to be clear about compatibility. What you test in your job or -- I apologize. What you define the requirements for in your job is whether Android devices will be compatible with Android, correct? A. Uhm, that seems like kind of a insufficiently precise way to put it. Q. Tell me the precise way that you would put it. A. What I would say that we do is, the compatibility program, in general, is intended to make sure that compatible Android devices can run applications written to the Android SDK. Q. Now, android does not support Java applications, correct? A. That is correct. Q. And so Android is not Java compatible, correct? A. That's correct." (Trial Phase I (Copyright) Vol 05 (Morrill), 1009:19-1010:7, Apr. 20, 2012)

197. Google's decision to create its own VM (Dalvik) means that Android is not fully Java compatible. For example, a January 22, 2008 email from Android engineer Dan Morrill states: "Please do not say that Android "is Java" or "runs Java". That is untrue; the Dalvik VM is not Java-compatible." GOOGLE-17-00113234.

[100] This can be seen by a visual inspection and comparison of https://docs.oracle.com/javase/1.5.0/docs/api/ and http://developer.android.com/reference/packages.html.
[101] This can be seen by a visual inspection and comparison of https://docs.oracle.com/javase/1.5.0/docs/api/ and http://developer.android.com/reference/packages.html.

198. Google engineers discussed that they wanted to use enough of the Java platform that they were "providing a familiar environment for our third-party developers" but that they were not attempting to be fully compatible with Java and they did not plan to "pass the TCK." GOOGLE-38-00015416-17; GOOGLE-24-00017719

199. There is evidence from Google that Java programs will not run on Android.

200. Google's expert witness Prof. Astrachan admitted in testimony that a program that is prepared to run on Java platform won't run on the Android platform, as the entry points are different:

> "Q. Now, Dr. Astrachan, you said that it was necessary to have the Method signatures in the Android API in order for this program to run, having been written for the Java APIs on Android; do you recall that testimony? A. Yes. Q. This program won't run on Android; will it, sir? And I emphasize the word "run." A. Running a program on Android is a different process than running a program on the Java Platform. Q. So a program that's prepared to run on the Java Platform won't run on the Android platform -- again emphasizing the word "run" -- correct, sir? A. The entry points are different on the platforms. Q. Can you answer my question "yes" or "no," sir? A. The program itself would need to be modified for the entry point so that it would run on the Android platform. Q. Isn't it true that the whole object or format in Android, the dex code in the Android, is different from the bytecode in Java, sir? A. I thought you were asking me about the code I wrote. I did not write dex code or bytecode. I wrote source code in the Java Programming Language. Q. Source code won't, quote, run, unquote, on anything; will it, sir? A. No. It has to be compiled to run. Q. And it has to be prepared to run on a platform; correct, sir? A. That is correct. Q. And so this program that you wrote on the board – Package Simple Author, et cetera -- it won't run until you do that preparation; correct, sir? A. Yes. You have to compile it to run on the platform, that is correct. Q. And so, and just to be clear, a program that has been compiled to run on a Java Platform will not run on an Android Platform because the format of the code in the Android Platform is different than the format of the code in the Java Platform. A. Can I understand your question to mean that the bytecode format and the dex code format, that's different. The source code is the same format, but the underlying code that runs on the platform is different." (Trial Phase I (Copyright) Vol 10 (Astrachan), 2210:12-2212:2, Apr. 27, 2012)

201. A senior Google engineer recently testified that the Dalvik and ART virtual machines in Android are only able to understand and execute Dalvik bytecode, not Java bytecode. (Ghuloum 30(b)(6), Anwar, 153:5-154:8, Dec. 9, 2015) He confirmed that it's not possible to run Java bytecode on either Dalvik or ART:

> "Q So Java bytecode and Dalvik bytecode in the Android context, those things are different, correct?... THE WITNESS: Yes, they are different. … Q And it's not possible to run Java bytecode directly on the Dalvik Virtual Machine or ART?... THE WITNESS: Today, it is not possible to do that." (Ghuloum 30(b)(6), Anwar, 154:21-155:6, Dec. 9, 2015) "Q Rather, what happens in the context of Dalvik or ART is that Java bytecode is converted first into Dalvik bytecode, and only at that point can the bytecode be executed by the Dalvik Virtual Machine or ART? A That's correct. Q Do you have any understanding why Google converts Java bytecode into Dalvik bytecode? … THE WITNESS: Yeah, I have some understanding of that. … Q And what's -- what's the

> reason for that? A So the -- the design of the -- of Dalvik bytecode and Java bytecode is -- is fairly different, and the reason for that is -- which gets at the heart of your question, is that the Dalvik bytecode was designed to be interpreted very fast and efficiently and to be much more compact than Java bytecode. So, for example, find that a Dalvik bytecode file is about as small -- I haven't checked lately, but is about as small as a compressed Java class file. Q Okay. Is it true that you could not take a compiled Java application and just run it directly on Dalvik or ART?... THE WITNESS: Yeah, that is true. By -- where by running directly, you mean -- I presume you mean the class file is consumed by the runtime and then executed… I assume you mean that the class file is consumed directly by the runtime and executed? … Q Correct. So is it true that a Java class file cannot be directly consumed by the Dalvik or ART Runtime and then be executed? A That's correct." (Ghuloum 30(b)(6), Anwar, 155:8-157:3, Dec. 9, 2015)

202. He also testified that a Java application cannot be assured to run on Android:

> "Q But it's true that if a developer – a Java developer goes to the Oracle website, downloads the Oracle JDK -- You're familiar with the Oracle JDK? A Yes. Q The Oracle JDK is -- is Oracle's official Java development kit, correct? A I believe so, yes. Q And so if a developer decides to download the JDK from Oracle and create an application -- with my assumption so far? A Yes. Q And they, in fact, create an application using the JDK, it's true that that application would not immediately and automatically run on Android?... THE WITNESS: If it's a -- not for all applications. Again, I think if it's something that is, for example, you know, a text-based application that is run from a command line, it would work probably perfectly fine on Android. It depends on what libraries that they're using. … Q Right. So if a developer over in the Android ecosystem uses JDK and they use libraries that are simply not present in the different Android class libraries, that developer's application will not run in Android? A That could be true, yes. There are cases where that would be true." (Ghuloum 30(b)(6), Anwar, 134:18-135:23, Dec. 9, 2015)

203. Further, evidence from Google establishes that Android applications do not run on Java and that Google does not take any steps to ensure that Android passes the Java compatibility tests (the TCK).

204. Dan Bornstein, a key Android developer, testified as follows:

> "Q. Would a Java Virtual Machine be available -- able to run the Dalvik core libraries if they were compiled to classfiles? A. As a general statement, no." (Bornstein, Daniel (Vol. 01) - 05/16/2011, 67:24-68:2, May 16, 2011)

205. Recently, an Android technical executive testified that Google does not attempt to ensure compatibility between Java and Android:

> "Q Okay. So Google has created its own Compatibility Test Suite for Android, correct? A Yes, that's correct. Q And Google does not use the Java TCK, in other words, the Oracle compatibility test in order to test Android? A That's correct. Q Those are different tests; the Oracle TCK test of compatability is a different test of compatability than the Android Compatability Test Suite? A Yes, that's correct. Q Are there any tests that Google performs to ensure that Java bytecode maps uniquely to DEX bytecode?... THE WITNESS: It's not -- it's not necessarily a goal of ours that Java code maps uniquely to

> DEX bytecode, meaning there's only one DEX bytecode incarnation of it. In fact, in the design -- the original design of Dalvik, it's -- it's specifically the case that that's, you know -- we could do optimization of DEX bytecode, so you could end up with multiple forms of DEX bytecode for a given thing. … Q So -- so the result of converting Java bytecode to DEX bytecode may be a situation in which there's not -- not a unique mapping between Java bytecode to DEX bytecode? A That's correct. Q Do you think it's possible that in the -- in the conversion of Java bytecode to DEX bytecode that some Java bytecode might just simply be ignored by -- by Android?... THE WITNESS: Well, there is -- there is bytecode that is ignored specifically by that compiler, because we don't support that bytecode. … Q Well, what do you mean, there's bytecode that is ignored specifically by which compiler? A By the DEX compiler that you're referring to." (Ghuloum 30(b)(6), Anwar, 188:15-192:12, Dec. 9, 2015)

206. On June 6, 2007, in an email from Andy Rubin, he recognized that: "[Sun] will never certify our VM, or put another way, I will not submit Dalvik for certification. Is it possible to certify without giving Sun access to the whole stack?" TX 246, GOOGLE-02-00089698-99.

207. As of December 2015, a high ranking Android executive with responsibilities for testing Android compatibility testified that he had never even looked at the Java TCK:

> "Q Do you have a role in dealing with compatibility issues -- compatibility issues in Android? A Yep. Q And notwithstanding that that's your role, you've never looked at the Java TCK? A Nope. Only look at the Java language specification. Q Okay. And so no -- anybody in your organization who deals with Android compatibility ever looked at or used the Java TCK? A Not that I'm aware of." (Ghuloum 30(b)(6), Anwar, 196:6-196:18, Dec. 9, 2015)

C. Google's Use of the 37 Java API Packages in Android Creates Fragmentation in the Java Developer Community

208. Google's unauthorized copying of the 37 API packages, its violation of the principles of no "subsetting" or "supersetting" in relation to the API packages that it copied from the official Java SE 5, Java SE 6 and Java SE 7 API specifications causes fragmentation to the Java platform and ecosystem. Google both subsetted and supersetted or otherwise included public or protected packages, classes, interfaces and fields other than those required or authorized by the Java API Specification at issue. I have discussed some examples in my report and Prof. Schmidt provides other examples in his report, which I incorporate and rely upon. (Schmidt Report, ¶ 105-107). Google also caused fragmentation by using the copied packages in compiled class libraries that are incompatible with Java, cause fragmentation to the Java platform and ecosystem. Fragmentation is a situation where (1) a Java developer writes an application for a particular Java specification (for example, a version of Java SE or Java ME), (2) expects to be able to use the API packages defined in the Java specification and expects their resultant application will run across devices that support Java according to the defined specification, but (3) then is faced with a Java-based platform that does not conform to the known and expected specifications.

209. Android is precisely such a platform, as it does not comply with any Java specification (Java SE, ME or any other) and, as discussed, does not pass Java's TCK compatibility tests. If a developer has created an application for Java SE 5, for example, using the API packages from Java, including the 37 packages at issue, he should be able to expect that the application could run on Android, given that Android uses 37 of the packages from Java SE 5. But, the application will not run on Android, and this breaks the developer's expectations regarding Java and its "write once, run anywhere" principle. The developer would have to expend cost and time redeveloping the Java application that met a particular Java specification, in order to get it to run on an incompatible Java platform such as Android, that does not meet the specification.[102]

210. For example, a developer writing an application and who had been using the full set of official Java API packages to do so, may write a program using packages such as javax.swing.[103] The developer may expect that in doing so, his program would run on a platform that supports the Java API packages.[104] However, for example, if the developer developed a program using API packages that are not in Android (e.g., javax.swing), the program will not run on Android. This is an example of fragmentation, from the lack of compatibility and interoperability between Java and Android, due to Google's impermissible copying of a "subset" of the Java APIs.

211. Consequently, Java application developers who historically could rely on the "write once, run anywhere" principle of Java, would have to expend time and energy writing and testing code for varied platforms, and learn multiple execution environments and support tools, which burdens them. If the Java developers were attracted to Android initially based on their familiarity with the expression of some of the

---

[102] Google has admitted that fragmentation of a development platform is harmful. In particular, Google has admitted that the same type of fragmentation that it caused to the Java ecosystem, is a harmful phenomenon within the Android ecosystem and that Google attempts to prevent such fragmentation within Android. For example, Android executive Anware Ghuloum testified: "Q Have you ever heard of the concept of fragmentation? A Yes, in many different contexts. Q Does Google worry about fragmentation of Android? A Yeah, that's a concern. Q Why is fragmentation of Android a concern to Google? A We would like it if developers targeting Android can ensure that with their applications, they're able to run on any Android implementation." (Ghuloum 30(b)(6), Anwar, 193:13-193:23, Dec. 9, 2015)

[103] Google both copied subsets of packages overall, but further also copied subsets of classes within a given package. For example, Google copied java.awt.font, but failed to support the following classes within java.awt.font: FontRenderContext, GlyphJustificationInfo, GlyphMetrics, GlyphVector, GraphicAttribute, ImageGraphicAttribute, LayoutPath, LineBreakMeasurer, LineMetrics, ShapeGraphicAttribute, TextHitInfo, TextLayout, TextLayout.CaretPolicy, TextMeasurer, and TransformAttribute. Similarly, in Android "Gingerbread" Google changed java.lang.System.java in a way that turned off java.security.SecurityManager.java, thus changing the expected behavior within this API package.

[104] For example, Google represents to Android application developers that Android contains "a set of core libraries that provides **most** [emphasis added] of the functionality available in the core libraries of the Java programming language." (Android Developer website, available at https://web.archive.org/web/20090228170145/http://developer.android.com/guide/basics/what-is-android.html.). Therefore, Java developers developing for the Java platform may believe that their Java applications will run on Android. Elsewhere in this report, I discuss that such developers are likely to be attracted to Android based on their familiarity with the Java API packages that *are* made available in Android.

API packages in Android, and discovered during the development process that they would have to make an investment in dealing with the subsetting or supersetting, having incurred that cost, there is a risk that they may move away from developing for Java over time and instead develop for Android.[105]

212. Also, the fragmentation of Java means that device manufacturers, such as phone makers, must support multiple divergent platforms. Consumer expectations may be disrupted as well, in that if they have a Java application, they will be less assured that it will be available for or be able to run on multiple devices and they would have to download particular applications for particular devices.

213. Google was aware when it first decided to base Android on Java; that Sun (and now Oracle) worked hard to minimize any potential for Java fragmentation. For example, April 13, 2006 emails between Android engineer Dan Bornstein and Android head Andy Rubin state: "We need to provide an alternative to MSFT, and we need to do it in such a way as we don't fragment 3rd party developers. … Java has very little fragmentation." TX 21, GOOGLE-02-00111218. Google was also aware that Sun required that any potential use of Java by Google maintain compatibility and not fragment Java. For example, [redacted] PX035, OAGOOGLE0100072596-98.

214. After Android was announced, industry observers commented on the problem of fragmentation that Android posed to Java (November 28, 2007, OAGOOGLE0100405426-28:

[105] Dan Bornstein testified that a programmer developing in a platform that was based on Java packages, but not containing everything in the official Java packages "would sometimes have called it an unpleasant surprise when they would find out that there was some other class or other thing they expected that wasn't present." (Bornstein, Daniel (Vol. 01) - 05/16/2011, 30:23-31:12, May 16, 2011)

> To put it bluntly: Android as it is currently defined is a fork of the Java ME platform. Android is similar to the Java ME, but it's a non-conformant implementation. Android is not compliant with Java ME nor is it compliant with Java SE. In fact, it's not really Java. Although it uses the Java programming language, the core APIs and the virtual machine are not consistent with the Java ME or SE platform - it's a fork. This was first pointed out by Stefano Mazzocchi in his November 12th blog entry entitled "Dalvik: how Google routed around Sun's IP-based licensing restrictions on Java ME". Stefano missed the fact that Android does not properly implement the CDC or CLDC Java ME APIs (a minimum requirement for Java ME conformance) - but kudos to him for being the first to report on the fork. The fork has since been picked up in the blogsphere by others here, here and elsewhere.

215. Other industry observers have noted the same. OAGOOGLE0000191668 (discussing "complaints about fragmentation" in Android, and observing that there would have been benefits if Google were "using/licensing" Java); OAGOOGL0004381807 ("it looks as though Google's Android is already beginning to fracture the Java mobile community."). Sun was concerned about Android because "API fragmentation means WORA goes out the window, which means the cost of exit goes way up and Sun's in trouble." OAGOOGLE0003997531-OAGOOGLE0003997532

216. There is evidence that Google was and continues to be aware that its incompatible use of the 37 Java APIs in Android fragmented Java. For example, on May 7, 2009, Andy Rubin referred to Android as a "fork" of Java. TX 172, GOOGLE-01-00029329.

D. The 37 API Packages in Android Are Technically Incompatible with the Java Platform

217. I have worked with Prof. Doug Schmidt who carried out analysis of and generated a report detailing numerous ways in which Android is incompatible with Java (in particular as relating to the 37 Java API packages). I rely upon and incorporate by reference in its entirety and refer to the contents of Prof. Schmidt's report. As detailed further in Prof. Schmidt's report, there are numerous reasons why Android is incompatible with Java. This technical incompatibility creates an incompatibility with Java developers' expectations that they can "write once, run anywhere" if they write programs targeted for a particular Java specification or platform.

218. [redacted] (Schmidt Report, ¶¶ 98, 107). [redacted] (Schmidt Report, ¶¶ 104). [redacted] (Schmidt Report, ¶¶ 103).

219. In general, Dex files from Android do not run on the Java Virtual Machine. Similarly, Class files from Java do not run on the Dalvik or ART virtual machines. (Schmidt Report, ¶¶ 11, 114, 117)

220. For these reasons, developers cannot be assured of compatibility, and Google's use of the Java API packages break the proposition of "write once, run anywhere."

VIII. CONCLUSIONS

221. Google copied thousands of lines of code and the structure, sequence and organization of the Java APIs without a license to do so. These Java APIs are valuable, copyrighted creative works.

222. The code and the structure, sequence and organization that Google copied is valuable to Google and the Android platform. Google faced market pressure to develop a platform quickly. The copied Java APIs provided pre-written programs that enabled Google to more quickly achieve adoption of Android. The copied Java APIs had been developed, industry-tested, and enhanced over the years, and so reflected a set of high quality resources. A very large community of trained developers had been created and nurtured over multiple years, providing a ready asset available to be deployed on new development. Usage of the copied Java APIs helped Android to become a stable platform in a relatively short amount of time.

223. The copied Java APIs are a critical and central resource within Android. Android will not function without the copied Java APIs, or even without the subset of the specifically copied lines of code, and there is a significant level of dependency on these copied APIs within Android. In addition, all of the most popular consumer apps that have been developed to run on Android also are heavily dependent on the 37 copied Java APIs.

224. Google viewed the open source licensing options for Java as unacceptable.

225. Google's copying of the Java APIs can be expected to have negatively impacted Oracle's business in both direct and indirect ways. The consistency of the Java APIs has been and continues to be a valuable asset that Oracle and Sun protected and promoted through its investments in both licensing and testing. Oracle received no direct licensing revenue nor software contributions from Google. Google admits, and independent testing confirms, that Android is incompatible with Java on multiple levels. This incompatibility reduces the size of the Java 'network', including its community of developers who are fragmented relative to what would be the case had Android been a licensed, compatible product.

TABLE OF APPENDICES

APPENDIX A – Glossary of terms

Acronym	Meaning	Description
CDC	Connected Device Configuration	▪ Configuration of Java ME for high end personal consumer and embedded devices ▪ Devices: Smart communicators, pagers, high-end PDAs, set top boxes[106]
CLDC	Connected Limited Device Configuration	▪ Sun's implementation / configuration of Java ME, designed for resource-constrained devices (non-smartphones)[107] ▪ Devices: Mobile phones, pagers, mainstream PDAs[108]
GPLv2	GNU General Public License, version 2, with classpath exception	Version of the GNU General Public License, released in 1991[109]
J2ME	Java Platform Micro Edition	Same as JavaME[110]
J2SE	Java 2 Standard Edition	▪ Now: Java SE ▪ Java SE was called J2SE between versions 1.2-1.5[111]
Java ME	Java Platform Micro Edition	▪ Allows applications to run on mobile devices: micro-controllers, sensors, gateways, mobile phones, PDAs, TV set top boxes, printers, etc.…[112]
Java SE	Java Platform Standard Edition	▪ Allows users to develop and deploy Java applications on desktops and servers and embedded environments[113]
JCP	Java Community Process	▪ Allows worldwide Java community to participate in the future of Java & APIs ▪ Established in 1998 ▪ The JCP is used in standardizing certain Java APIs in feature phones and defragment the Java ME market such that handset makers don't have to make handsets that run the software of specific software makers—they can download the standard Java APIs that'll do that

[106] *See, e.g.*, Java ME Technology – CDC, Oracle, http://www.oracle.com/technetwork/java/javame/tech/index-jsp-139293.html (last visited Jan. 5, 2016).
[107] *See, e.g.*, Connected Limited Device Configuration (CLDC); JSR 139, Oracle, http://www.oracle.com/technetwork/java/cldc-141990.html (last visited Jan. 5, 2016).
[108] *See, e.g.*, Connected Limited Device Configuration (CLDC); JSR 139, Oracle, http://www.oracle.com/technetwork/java/cldc-141990.html (last visited Jan. 5, 2016).
[109] gnu general public license, version 2, GNU Operating System, http://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html (last visited Jan 5, 2016).
[110] what is j2me or java me?, Oracle, https://java.com/en/download/faq/whatis_j2me.xml (last visited Jan 5, 2016).
[111] j2se 1.4.2, Oracle, http://www.oracle.com/technetwork/java/javase/index-jsp-138567.html (last visited Jan 5, 2016).
[112] java platform, micro edition (java me), Oracle, http://www.oracle.com/technetwork/java/embedded/javame/index.html (last visited Jan 5, 2016).
[113] java se at a glance, Oracle, http://www.oracle.com/technetwork/java/javase/overview/index.html (last visited Jan 5, 2016).

Acronym	Meaning	Description
JDK	Java Development Kit	▪ Allows the writing of Java applications ▪ Includes: Java Runtime Environment, Java Compiler and the Java APIs[114]
JRE	Java Runtime Environment	▪ What you get when you download Java software ▪ Includes: JVM, Java platform core classes, supporting Java libraries[115]
JSR	Java Specification Requests	▪ Basically requests for specific changes to the Java language, libraries, and other components. JSRs are created as part of the Java Community Process, where interested parties can propose ideas for enhancements and the JCP executive committee will review and approve as appropriate ▪ They are APIs like Mobile 3D Graphics (JSR 184), Bluetooth (JSR 82), PDA Optional Packages (JSR 75), Wireless Messaging (JSR 205), Mobile Media (JSR 135), Advanced Multimedia Supplements (JSR 234) [116]
JVM	Java Virtual Machine	▪ Works with/on top of operating systems to run Java applications ▪ What allows Java programs to run on Windows & Mac[117]
MIDP	Mobile Information Device Profile	▪ MIDP is connected with CDC to create the "runtime environment" for mobile devices ▪ Provides core application functionality for mobile devices ▪ Asko Komsi (Nokia's Director of Industry Relations): In the early days of J2ME, MIDP was one of the APIs, nobody defined a framework. Carriers offered MIDP support on many handsets offered[118]
OpenJDK	Open Java Development Kit	An open-source implementation of Java SE. Companies and individuals license through the GPL model[119]
TCK	Technology Compatibility Kit	Test suite to show compatibility with Java[120]

[114] *See, e.g.*, Java™ Platform Standard Edition 7 Names and Versions, Oracle, http://www.oracle.com/technetwork/java/javase/jdk7-naming-418744.html (last visited Jan. 5, 2016).
[115] Java Programming Environment and the Java Runtime Environment (JRE), Oracle, https://docs.oracle.com/cd/E19455-01/806-3461/6jck06gqd/index.html (Last visited Jan. 05, 2016).
[116] *See, e.g.*, JSR Overview, Java Community Process, https://jcp.org/en/jsr/overview (last visited Jan. 5, 2016).
[117] java virtual machine, Business Dictionary, http://www.businessdictionary.com/definition/Java-virtual-machine-JVM.html (last visited Jan 5, 2016).
[118] Mobile Information Device Profile (MIDP); JSR 118, Oracle, http://www.oracle.com/technetwork/java/index-jsp-138820.html (last visited Jan. 5, 2016).
[119] OpenJDK, http://openjdk.java.net/ (Last visited Jan. 05, 2016).
[120] TCK Project Planning and Development Guide, (August, 2003), at pg. 2 https://jcp.org/aboutJava/communityprocess/ec-public/TCK-docs/ppg.pdf

APPENDIX B - CV

Curriculum Vitae

CHRIS F. KEMERER

EDUCATION

The Wharton School, University of Pennsylvania

B.S., *magna cum laude,* 1979. Dual Major: Decision Sciences and Economics. Dean's List every semester. Alcoa Foundation Scholarship, General Mills Leadership Scholarship, National Honor Society Scholarship.

University of Edinburgh, Edinburgh, Scotland, U.K.

Selected to represent the University of Pennsylvania in the inaugural Penn-Edinburgh Exchange Program, academic year 1976-77. Course work included Computer Science, Economic and Social History. Merit certificates in all subjects.

Carnegie Mellon University

Ph.D., 1987, Systems Sciences (Information Systems), Graduate School of Industrial Administration. Minor in Accounting. **M.S.** 1986, Systems Sciences. NL Industries Fellowship, 1983-84. William Larimer Mellon Fellowship, 1984-85. IBM Fellowship, 1985-86. ICIS Doctoral Fellow, 1985.

Dissertation title: "Measurement of Software Development Productivity"

ACADEMIC APPOINTMENTS

MIT, Sloan School of Management (1987-1995)

Assistant Professor of Management Science (1987-1989)

Douglas Drane Career Development Assistant Professor of Information Technology and Management (1989-1992)

Douglas Drane Career Development Associate Professor of Information Technology and Management (1992-1995)

Acting Associate Dean for Information Technology (1994-1995)

The Wharton School, University of Pennsylvania

Visiting Associate Professor, Operations and Information Management Department, Spring 1994

Katz Graduate School of Business, University of Pittsburgh (1995-present)

David M. Roderick Chair in Information Systems (1995-Present)

(joint appointment in **School of Information Sciences**, 1998-Present)

Carnegie Mellon University, School of Computer Science (1999-present)

Adjunct Professor of Software Engineering (1999-Present)

PROFESSIONAL EXPERIENCE

American Management Systems, Inc.,

June 1979 to August 1983 - Managed custom software development projects and provided consulting services to a variety of public and private sector clients for this management consulting and computer services firm based in metropolitan Washington, D.C. Youngest staff member to be promoted to *Principal* in the firm's then twelve year history.

RESEARCH

Research Interests

Management of Information Systems

Software Engineering Measurement and Modeling

Accounting and Economic Issues in Information Systems

Technology adoption and diffusion

Researcher Ranking (International)

Ranked Top 12 in the world in authorship of highly cited papers, by Iivari, J., "Making Sense of the History of Information Systems Research 1975-1999: A View of Highly Cited Papers", *Communications of the AIS*, v. 36, 2015.

INFORMS Information Systems Society *Distinguished Fellow*, inducted 2013.

"A Metrics Suite for Object Oriented Design", (with S. Chidamber) ranked in the Top 10 most highly cited IS research papers worldwide (1993-2010) by N. Hassan, "Who else is Reading Our Research? An Exploratory Scientometric Method of Assessing IS Relevance Outside the Allied Management and Computing Fields", Univ. of Minnesota working paper, 2011.

ISI/Thomson-Reuters "Highly Cited" Computing Researcher (International)

Ranked Top 50 Most Cited Author in top journals, 1990-2004, by Lowry, P.B., et al., "Assessing Leading Institutions, Faculty and Articles in Premier Information Systems Research Journals", *Communications of the AIS*, v. 20, 2007

Ranked Top 10 in the world in terms of research publications, by Athey S. and Plotnicki, J. "An Evaluation of Research Productivity in Academic IT", *Communication of the AIS*, v.3, March 2000.

Ranked Top 25 in the world in terms of research publications and productivity, by Im, K.S., Kim, K. Y., and Kim, J. S. "An Assessment of Individual and institutional research productivity in MIS", *Decision Line*, January 1998.

Research Publications (order of authorship is alphabetic on multi-authored articles unless otherwise shown)

Articles

"An Empirical Validation of Software Cost Estimation Models", *Communications of the ACM*, v. 30 n. 5, pp. 416-429, May 1987.

(Reprinted in *Software Engineering Metrics*, M. J. Shepperd (ed.), McGraw-Hill, 1994.)

(Highly Cited Paper, Iivari, Juhani (2015) "Making Sense of the History of Information Systems Research 1975-1999: A View of Highly Cited Papers," *Communications of the Association for Information Systems*: Vol. 36, Article 25.)

"Factors Affecting Software Maintenance Productivity: An Exploratory Study", (with R. Banker and S. Datar), *Proceedings of the 8th International Conference on Information Systems (ICIS)*, Pittsburgh, Pennsylvania, pp. 160-175, December 1987.

"Software Production Economics: Theoretical Models and Practical Tools", *ACM 27th Technical Symposium Proceedings*, Gaithersburg, Maryland, June 1988.

"Barriers to Successful Strategic Information Systems", (with G. Sosa), *Planning Review*, Special Issue on Strategic Information Systems, v. 16, n. 5, September / October 1988, pp. 20-46.

"Scale Economies in New Software Development" (with R. Banker), *IEEE Transactions on Software Engineering*, v. 15, n. 10, pp. 1199-1205, October 1989.

"An Agenda for Research in the Managerial Evaluation of Computer-Aided Software Engineering Tool Impacts", *Proceedings of the 22nd Hawaii International Conference on System Sciences,* v. II, pp. 219-228, January 1989.

"An Agent-theoretic Perspective on the Management of Information Systems", (with V. Gurbaxani), *Proceedings of the 22nd Hawaii International Conference on System Sciences*, v. III, pp. 141-150, January 1989.

(Awarded "Best Paper", Economics of Information Systems Management Minitrack.)

"Instrumenti per la Rilevazione ed il Recupero della Produttivita' nella Manutenzione del Software" in *Sistemi & Impresa* (Italian), v. 2, March 1990. Translated from "Tools for Measuring and Improving Software Maintenance Productivity: Results from Empirical Research on Software Complexity" *Proceedings of the Software Engineering Symposium 90*. Milan, Italy, February 1990.

"A Quantitative Analysis of US and Japanese Practice and Performance in Software Development", (with M. Cusumano), *Management Science*, v. 36, n. 11, pp. 1384-1406, November 1990.

"An Agency Theory View of the Management of End-User Computing", (with V. Gurbaxani), *Proceedings of the 11th International Conference on Information Systems (ICIS)*, Copenhagen, Denmark, pp. 279-289, December 1990.

"A Model to Evaluate Variables Impacting Productivity on Software Maintenance Projects", (with R. Banker and S. Datar), *Management Science*, v. 37, n. 1, pp. 1-18, January 1991.

("Top 100" Most Cited Article, 1990-2004, Lowry, P.B., *et al*., "Assessing Leading Institutions, Faculty and Articles in Premier Information Systems Research Journals", *Communications of the AIS*, v. 20, 2007)

"Systems Development Risks in Strategic Information Systems", (with G. Sosa), *Information and Software Technology*, v. 33, n. 3, pp. 212-223, April 1991.

(Translated into Dutch and reprinted as "Risico's bij systeemontwikkeling van strategische informatiesystemen" in *Management en Organisatie van Automatiseringsmiddelen*, Kluwer Bedrijfswetenschappen, Netherlands, October 1993).

"Software Cost Estimation Models", Chapter 28 in *The Software Engineer's Reference Book*, Butterworth-Heinemann Ltd., Oxford, England, UK. (1991).

"Towards a Suite of Object-Oriented Software Design Metrics", (with S. Chidamber), *Proceedings of the Sixth ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA),* Phoenix, Arizona, pp. 197-211, October, 1991.

(Published as *ACM SigPlan Notices* v. 26, n. 11, November, 1991)

"Cyclomatic Complexity Density and Software Maintenance Productivity", (with G. Gill), *IEEE Transactions on Software Engineering*, v. 17, n. 12, pp. 1284-1288, December 1991.

"How the Learning Curve Affects Computer Aided Software Engineering (CASE) Tool Adoption", *IEEE Software*, v. 9, n. 5, pp. 23-28, May 1992.

(Nominated for "Best Article of the Year, 1992", *IEEE Software* editorial board.)

(Reprinted in *Computer-Aided Software Engineering (CASE),* E. Chikofsky (ed.), 2nd edition, pp. 154-159, IEEE Computer Society Press, 1993.)

"Object-Oriented and Conventional Analysis and Design Methodologies: Comparison and Critique" (with R. Fichman), *IEEE Computer*, v. 25, n. 10, pp. 22-39, October 1992.

(Reprinted in *Readings in Object-Oriented Systems and Applications*, D. Rine (ed.), IEEE Computer Society Press, 1993.)

(Reprinted in *High-Integrity System Specification and Design*, J. Bowen and M. Hinchey, (eds.), Springer-Verlag, Berlin, 2000.)

"Improving the Reliability of Function Point Measurement: An Empirical Study", (with B. Porter) *IEEE Transactions on Software Engineering*, v. 18, n. 10, pp. 1011-1024, November 1992.

"Recent Applications of Economic Theory in Information Technology Research" (with J. Y. Bakos), *Decision Support Systems*, v. 8, n. 5, pp. 365-386, December 1992.

"Performance Evaluation Metrics for Information Systems Development: A Principal-Agent Model", (with R. Banker) *Information Systems Research*, v. 3, n. 4, pp. 379-400, December 1992.

"Staffing Factors in Software Cost Estimation Models" (with M. Patrick) Chapter 11 in the *Handbook of Software Engineering Productivity*, J. Keyes, (ed.), McGraw-Hill, New York, NY, pp. 175-190, 1993.

"Bridging the Gap between Research and Practice in Software Engineering Management: Reflections on the Staffing Factors Paradox", pp. 116-120 in *Experimental Software Engineering Issues: Critical Assessment and Future Directions*, Rombach, H. D., V. R. Basili and R. W. Selby (eds.), Springer-Verlag, Berlin, (1993).

"Adoption of Software Engineering Process Innovations: The Case of Object-Orientation" (with R. Fichman), *Sloan Management Review*, vol. 34, n. 2, pp. 7-22, Winter 1993.

(Translated into Dutch and reprinted as "Acceptatie van programmeertechnische procesinnovaties: de objectorièntatie" in *Management en Organisatie van Automatiseringsmiddelen*, Kluwer Bedrijfswetenschappen, Netherlands, March 1995).

"Reliability of Function Points Measurement: A Field Experiment", *Communications of the ACM*, v. 36, n. 2, pp. 85-97, February 1993.

"Toward a Theory of the Adoption and Diffusion of Software Process Innovations", (with R. Fichman) *IFIP Working Conference on Diffusion, Transfer and Implementation of Information Technology,* Pittsburgh, Pennsylvania, October 1993.

"Software Complexity and Software Maintenance Costs", (with R. Banker, S. Datar and D. Zweig), *Communications of the ACM*, v. 36, n. 11, pp. 81-94, November 1993.

"Evidence on Economies of Scale in Software Development", (with R. Banker, and H. Chang), *Information and Software Technology*, v. 36, n. 5, pp. 275-282, 1994.

"A Metrics Suite for Object Oriented Design", (with S. Chidamber), *IEEE Transactions on Software Engineering*, v. 20, n. 6, pp. 476-493, June 1994.

(Highly Cited Paper, Iivari, Juhani (2015) "Making Sense of the History of Information Systems Research 1975-1999: A View of Highly Cited Papers," *Communications of the Association for Information Systems*: Vol. 36, Article 25.)

"Computerized Loan Origination Systems: An Industry Case Study of the Electronic Markets Hypothesis", (with C. Hess), *MIS Quarterly*, v. 18, n. 3, pp. 251-275, September 1994.

(Awarded "Best Paper of 1994" by the *MIS Quarterly* editors)

"Software Complexity and Software Maintenance: A Survey of Empirical Research", *Annals of Software Engineering*, v. 1, n. 1, pp. 1-22, August 1995.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", (with E. Brynjolfsson) *Management Science*, v. 42, n. 12, pp. 1627-1647, December 1996.

(Highly Cited Paper, Iivari, Juhani (2015) "Making Sense of the History of Information Systems Research 1975-1999: A View of Highly Cited Papers," *Communications of the Association for Information Systems*: Vol. 36, Article 25.)

"A Longitudinal Analysis of Software Maintenance Patterns", (with S. Slaughter) *18th International Conference on Information Systems*, research in progress, Atlanta, Georgia, December 1997.

"Methodologies for Performing Empirical Studies: Report from the International Workshop on Empirical Studies of Software Maintenance" (with S. Slaughter) *Empirical Software Engineering*, v. 2, n. 2, pp. 109-118, 1997.

"The Assimilation of Software Process Innovations: An Organizational Learning Perspective", (with R. Fichman) *Management Science*, v. 43, n. 10, pp. 1345-1363, October 1997.

("Top 100" Most Cited Article, 1990-2004, Lowry, P.B., *et al*., "Assessing Leading Institutions, Faculty and Articles in Premier Information Systems Research Journals", *Communications of the AIS*, v. 20, 2007)

(Highly Cited Paper, Iivari, Juhani (2015) "Making Sense of the History of Information Systems Research 1975-1999: A View of Highly Cited Papers," *Communications of the Association for Information Systems*: Vol. 36, Article 25.)

"Determinants of Software Maintenance Profiles: An Empirical Investigation" (with S. Slaughter) *Journal of Software Maintenance*, v. 9, pp. 235-251, 1997.

"Object Technology and Software Reuse: Lessons from Longitudinal Case Studies of Early Adopters", (with R. Fichman) *IEEE Computer*, v. 30, n. 10, pp.-47-59, October 1997.

(Translated into Japanese and reprinted in *Nikkei Computer*, pp. 145-153, April 13, 1998).

(Translated into Japanese and reprinted in a 'Best of' collection of papers, *Nikkei Computer Books*, ISBN 4-8222-0771-4, 1998.)

"Future Markets: Information Technology's Impact on Market Structure" Chapter 1 in *Information Technology and Industrial Competitiveness: How IT Shapes Competition*, C. F. Kemerer (ed.), Kluwer Academic Press, 1998.

"A Longitudinal Empirical Analysis of Software Evolution", (with S. Slaughter), *International Workshop on Principles of Software Evolution*, Kyoto, Japan, April 20-21, 1998.

"Progress, Obstacles, and Opportunities in Software Engineering Economics", *Communications of the ACM*, v. 41, n. 8, pp. 63-66, August 1998

"Managerial Use of Object Oriented Software Metrics", (with S. Chidamber and D. Darcy), *IEEE Transactions on Software Engineering*, v. 24, n. 8, pp. 629-639, August 1998.

(Highly Cited Paper, Iivari, Juhani (2015) "Making Sense of the History of Information Systems Research 1975-1999: A View of Highly Cited Papers," *Communications of the Association for Information Systems*: Vol. 36, Article 25.)

"Toward a Detailed Classification Scheme for Software Maintenance Activities", (with E. Barry and S. Slaughter), *Proc. of the 5th Americas Conference on Information Systems*, August 1999.

(Best paper award, Software Process Improvement mini-track)

"An Empirical Approach to Studying Software Evolution" (with S. Slaughter), *IEEE Transactions on Software Engineering*, v. 25, n. 4, pp. 493-509, July/August 1999.

"The Illusory Diffusion of Innovations: An Examination of Assimilation Gaps", (with R. Fichman), *Information Systems Research*, v. 10, n. 3, pp. 255-275, September 1999.

(Highly Cited Paper, Iivari, Juhani (2015) "Making Sense of the History of Information Systems Research 1975-1999: A View of Highly Cited Papers," *Communications of the Association for Information Systems*: Vol. 36, Article 25.)

"An Empirical Analysis of Software Evolution Profiles and Outcomes", (with E. Barry and S. Slaughter), *Proc. of the 20th International Conference on Information Systems*, December 1999.

Bennett, K. *et al.*, "Empirical studies of evolving systems", *Empirical Software Engineering*, v. 4, n. 4, December 1999.

"Incentive Compatibility and Systematic Software Reuse ", (with R. Fichman), *Journal of Systems and Software*, v. 57, n. 1, pp. 45-60, April 2001.

"Software Errors and Software Maintenance Management", (with R. Banker, S. Datar, and D. Zweig), *Information Technology and Management*, v. 3, nos. 1/2, pp. 25-41, January 2002.

"Activity Based Costing for Component-based Software Development ", (with R. Fichman), *Information Technology and Management*, v. 3, nos. 1/2, pp. 137-160, January 2002.

Ives, B. et al., "What Every Business Student Needs to Know About Information Systems," *Communications of the Association for Information Systems*, v. 9, n. 30, 2002.

"On the Uniformity of Software Evolution Patterns", (with E. Barry and S. Slaughter), *Proc. of the 25th International Conference on Software Engineering*, Portland, Oregon, May 2003.

MacCormack, A., C. Kemerer, M. Cusumano, and B. Crandall, "Exploring trade-offs between productivity and quality in the selection of software development practices" *IEEE Software*, v. 20, n. 5, pp. 78-85, September/October 2003.

Cusumano, M, MacCormack, A., Kemerer, C., Crandall, B., "Software Development Worldwide: The State of the Practice", *IEEE Software,* v. 20, n. 6, pp. 28-34, November/December 2003.

(Selected as one of *IEEE Software*'s 25th Anniversary's best peer-reviewed full length articles: "*IEEE Software*'s 25th Anniversary Top Picks", *IEEE Software*, January/February 2009, pp. 9-11.)

Ramasubbu, N., Mithas S., Krishnan M.S, and C. Kemerer, "Empirical Analysis of Quality Management Practices in Distributed Custom Software Development" *Proc. of the Academy of Management Annual Meeting*, New Orleans, LA, August 2004.

"OO Metrics in Practice", (with D. Darcy), *IEEE Software*, v. 22, n. 6, pp. 17-19, November/December 2005.

"The structural complexity of software: An experimental test" (with D. Darcy, S. Slaughter, and J. Tomayko), *IEEE Transactions on Software Engineering*, v. 31, n. 11, pp. 982-995, November 2005.

"Environmental Volatility, Development Decisions and Software Product Volatility: A Longitudinal Study", (with E. Barry and S. Slaughter), *Management Science*, v. 52, n. 3, pp. 448-464, March 2006.

(Nominated for Alfred P. Sloan Foundation Annual Industry Studies Best Paper Prize, 2006.)

"How Software Process Automation Affects Software Evolution: A Longitudinal Empirical Analysis", (with E. Barry and S. Slaughter), *Journal of Software Maintenance and Evolution*, v. 19, n. 1, pp. 1-31, January-February 2007.

"Network Effects in the Flash Memory Market: A Preliminary Analysis" (with C. Liu, S. Slaughter, and M. Smith) *Workshop on Information Systems and Economics* (WISE), Montreal, Canada, December 2007.

Ramasubbu, N., Mithas S., Krishnan M.S, and C. Kemerer, "Work Dispersion, Process-Based Learning and Offshore Software Development Performance" *MIS Quarterly*, v. 32, n. 2, pp. 437-458, June 2008.

"The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data", (with M. Paulk), *IEEE Transactions on Software Engineering*, vol. 35, no. 4, pp. 534-550, July/Aug. 2009.

Cusumano, M, A. MacCormack, C. Kemerer, B. Crandall, "Critical Decisions in Software Development: Updating the State of the Practice", *IEEE Software*, pp. 84-87, September/October 2009.

(Translated into Japanese and reprinted in *Nihon Keizai Shinbunsha* (Japan Economic Newspaper) Electronics, Issue 1029, pp. 155-161, 2010.

Heck, Eric van, *et al.*, "Are We Wise about Sub-Fields in IS? Lessons from Forming and Sustaining a Research Community", Paper 187, *Proc. of the 30th International Conference on Information Systems*, December 2009.

Swanson, B., J. King, I. Benbasat, C. Kemerer, "The Early Years of ISR: Recollections of the Editors", *Information Systems Research*, v. 21, n. 4, pp. 665-674, December 2010.

Liu, C., E. Gal-Or, C. Kemerer, M. Smith, "Compatibility and Proprietary Standards: The Impact of Conversion Technologies in IT-Markets with Network Effects", *Information Systems Research*, v. 22, n. 1, pp. 188-207, March 2011.

Jiang, Y., J. Shang, C. Kemerer, and Y. Liu, "Optimizing E-tailer Profits and Customer Savings: **Pricing Multistage Customized Online Bundles**", *Marketing Science*, v. 30, pp. 737-752, July-August 2011.

Harter, D., C. Kemerer and S. Slaughter "Does Software Process Improvement Reduce the Severity of Defects? A Longitudinal Field Study", *IEEE Transactions on Software Engineering*, v. 38, n. 4, pp. 810-827, July/August 2012.

Liu, C., C. Kemerer, S. Slaughter, M. Smith, " Standards Competition In the Presence of Conversion Technology: An Empirical Analysis of the Flash Memory Card Market ", *MIS Quarterly*, v. 36, n. 3, pp. 921-942, September 2012.

Ramasubbu, N., C. Kemerer, and J. Hong, "Structural Complexity and Programmer Team Strategy: An Experimental Test", *IEEE Transactions on Software Engineering*, v. 38, n. 5, pp. 1054-1068, September/October 2012.

"Strategies for Tomorrow's "Winners-Take-Some" Digital Goods Markets", (with C. Liu and M. Smith), *Communications of the ACM*, v. 56, n. 5, pp. 76-82, May 2013.

Ramasubbu, N. and C. Kemerer, "Towards a Model for Optimizing Technical Debt in Software Products", *International Conference on Software Engineering* (*ICSE*) Workshop on Technical Debt, San Francisco, CA, May 2013.

Ramasubbu, N. and C. Kemerer, "Managing Technical Debt in Enterprise Software Packages", *IEEE Transactions on Software Engineering*, v. 40, n. 8, pp. 758-772, August 2014.

Ramasubbu, N., C. Kemerer, and J. Woodward, "Managing Technical Debt: Insights from Recent Empirical Evidence", *IEEE Software*, pp. 12-15, March/April 2015.

Ramasubbu, N. and C. Kemerer, "Technical Debt and the Reliability of Enterprise Software Systems: A Competing Risks Analysis", accepted for publication, *Management Science*, 2/15/2015.

Working papers

"Class Warfare: A Re-Examination of Video Game Console Competitions", (with B. K. Dunn), working paper, May 2012.

Cases

MacCormack, A, B. K. Dunn and C. Kemerer, "Research in Motion: The Mobile OS Platform Wars", *Harvard Business School Case* 613-001, June 2012.

MacCormack, A, B. K. Dunn and C. Kemerer, "Barnes & Noble: Managing the E-Book Revolution", *Harvard Business School Case* 613-073, January 2013.

Books

Software Project Management: Readings and Cases, (ed.) ISBN 0-256-18545-X, Richard D. Irwin/McGraw-Hill, Boston, MA, 1997.

Information Technology and Industrial Competitiveness: How Information Technology Shapes Competition, (ed.), ISBN 0-7923-8020-7, Kluwer Academic Publishers, Norwell, MA, 1998.

Conference Proceedings

Proceedings of the Sixteenth International Conference on Information Systems, (co-editor, with J. DeGross, G. Ariav, C. Beath, R. Hoyer, eds.) Amsterdam, the Netherlands, Association for Computing Machinery (ACM) Press, 1995.

Recent Research Grants

"Discovering the Profiles and Patterns of Software Evolution", *National Science Foundation* #9988315 & #9988227, co-Principal Investigator with S. Slaughter.

"Economics of Software Maintenance as part of Total Cost of Ownership", *IBM Faculty Awards Program Maximum Award.*

"Standards Competition in the Presence of Conversion Technology: An Empirical Analysis of the Flash Memory Card Market", (with C. Liu and M. Smith), *NET Institute, New York University.*

"Consumer Welfare in Two-Sided Digital Content Markets", *BNY Mellon Faculty Fellowship.*

Journalism

"Investigate All Options" *InformationWeek* April 27, 1998, p. 170.

"Value Your App Inventory" *InformationWeek* May 25, 1998, p. 130.

"Hidden Costs of New Technology" *InformationWeek*, June 15, 1998, p. 168.

"Factor in Cost Avoidance" " *InformationWeek*, August 3, 1998, p. 72.

"Maximize Reuse by Monitoring" *InformationWeek*, September 14, 1998, p. 390.

"Reusable Asset" *InformationWeek*, September 22, 1998, pp. 64-66.

"Winners Take Some", *Pitt Business*, Spring 2014, pp. 7-9.

Presentations

"Productivity in Software Maintenance", invited presentation at the *Nippon Univac Kaisha, Ltd. Advanced Technology Seminar*, Pittsburgh, Pennsylvania, September 1986. (Simultaneous translation, English-Japanese, Japanese-English).

"An Empirical Validation of Software Cost Estimation Models", *7th International Conference on Information Systems*, San Diego, California, December 1986.

"Factors Affecting Software Maintenance Productivity", invited presentation at the *UCLA Computers & Information Systems Colloquium*, Los Angeles, California, May 1987.

"An Empirical Validation of Software Cost Estimation Models", *Software Engineering Institute COCOMO User's Group Meeting*, Pittsburgh, Pennsylvania, November 1987.

"Cost Estimation for Software Management", invited presentation at the *Royal Aeronautical Society's Management of Major Software Projects Symposium,* London, England, UK, November 1987.

"Factors Affecting Software Maintenance Productivity: An Exploratory Study", *8th International Conference on Information Systems*, Pittsburgh, Pennsylvania, December 1987.

"Models for Software Development Management", invited presentation at the *University of Minnesota Management Information Systems Research Center*, Minneapolis, Minnesota, January 1988.

"Scale Economies in New Software Development", invited presentation at the *University of Minnesota Management Information Systems Workshop*, Minneapolis, Minnesota, January 1988.

"Models for Software Development Management", invited presentation at the Boston Chapter of the *Association for Systems Management*, Boston, Massachusetts, April 1988.

"Software Production Economics: Theoretical Models and Practical Tools", *27th ACM Technical Symposium*, Gaithersburg, Maryland, June 1988.

Panel member, "How do we evaluate the use of CASE?", *Second International Workshop on Computer-Aided Software Engineering*, Boston, Massachusetts, July 1988.

Panel member, "Modeling the Software Maintenance Process: I/O Summary Models" *IEEE Conference on Software Maintenance*, Phoenix, Arizona, October 1988.

Panel chair, "Debates on the Value of Agency Theory to Research in Management Information Systems" *9th International Conference on Information Systems*, Minneapolis, Minnesota, December 1988.

"An Agenda for Research in the Managerial Evaluation of Computer-Aided Software Engineering Tool Impacts", *22nd Hawaii International Conference on System Sciences*, Kailua-Kona, Hawaii, January 1989.

"An Agent-theoretic Perspective on the Management on Information Systems", with V. Gurbaxani, *22nd Hawaii International Conference on System Sciences*, Kailua-Kona, Hawaii, January 1989.

"Measurement and Modeling for Improved Software Maintenance Management", invited presentation, *Software Engineering Symposium 89*, Milan, Italy, January 1989. (Simultaneous translation, Italian-English, English-Italian).

"Effective Measurement and Modeling of Software Maintenance Productivity", invited presentation, *Annual Meeting and Conference of the Software Maintenance Association,* Atlanta, Georgia, May 1989.

"A Quantitative Analysis of Software Engineering Practice and Performance in the United States and Japan", *TIMS XXIX*, Osaka, Japan, July 1989.

"Research in Software Engineering Management", *International Financial Services Research Center*, London, England, UK, September 1989.

"Performance Evaluation of IS Development Manager-Agents", (with R. Banker), *First Workshop on Information Systems and Economics*, Cambridge, MA, December 1989.

Session chair, "Software Testing and Maintainability", *10th International Conference on Information Systems*, Boston, Massachusetts, December 1989.

"Tools for Measuring and Improving Maintenance Productivity", invited presentation, *Software Engineering Symposium 90*, Milan, Italy, February 1990. (Simultaneous translation, Italian-English, English-Italian).

"Function Point Measurement Reliability: Preliminary Results", Feature Speaker, *International Function Point User Group Spring Conference*, Lake Buena Vista, Florida, April 1990.

"Software Productivity and Complexity Modeling: An Empirical Study of Software Development and Maintenance Projects", *TIMS / ORSA 29th Joint National Meeting*, Las Vegas, Nevada, May 1990.

"Reliability of Function Points Measurement: A Field Experiment", invited presentation, *University of California-Irvine Information and Computer Science Colloquium Series*, Irvine, California, September 1990.

"Function Points Measurement Reliability: Results of an Empirical Study", Feature Speaker, *International Function Point Users Group Fall Conference*, San Antonio, Texas, October 1990.

"An Agency Theory View of the Management of End-User Computing", *11th International Conference on Information Systems*, Copenhagen, Denmark, December 1990.

"Software Development Measurement and Performance Evaluation", Invited Speaker, *Danish Technological Institute Information Technology Strategy Symposium*, Taastrup, Denmark, December 1990.

"Reliability of Function Points Measurement: A Field Experiment", invited presentation, *The Institute of Management Sciences, Boston Chapter Seminar*, Boston, MA, March 1991.

"Reliability of Function Points Measurement: A Field Experiment", invited presentation, *University of Rochester, William E. Simon Graduate School of Business Administration Computer and Information Systems Seminar*, Rochester, NY, April 1991.

"Measurement for Improved Software Development", invited presentation, *Credit Suisse Management Forum*, Zurich, Switzerland, May 1991.

"Measurement for Improved Software Development", invited presentation, *AT&T Bell Labs*, Naperville, IL, June 1991.

"Reliability of Function Points Measurement: A Field Experiment", invited presentation, *University of Texas IS Lectureship Series*, Austin, TX, October 1991.

"Performance Evaluation Metrics for Information Systems Development: A Principal-Agent Model", invited presentation, *ORSA/TIMS Joint National Meeting*, Anaheim, CA, November 1991.

"Reliability of Function Points Measurement: A Field Experiment", invited presentation, *University of Chicago Graduate School of Business Research Seminar*, Chicago, Illinois, November, 1991.

Panel chair, "Current Research on Economics and Information Systems: A Report from the WISE Preconference Meeting" *12th International Conference on Information Systems*, New York, NY, December 1991.

"Software Development Measurement and Performance Evaluation", invited presentation, *Banca d' Italia*, Rome, Italy, May 1992.

"Reliability of Function Points Measurement: A Field Experiment", *European Software Cost Modelling Meeting*, Munich, Germany, May 1992.

"Reliability of Function Points Measurement: A Field Experiment", *MITRE-Washington Economic Analysis Center Conference on Analytical Methods in Software Engineering Economics II*, McLean, Virginia, July 1992.

"Bridging the Gap between Research and Practice in Software Engineering Management: Reflections on the Staffing Factors Paradox", Invited position paper, *Dagstuhl Workshop on Experimental Software Engineering Issues*, Dagstuhl, Germany, September 1992

"Adoption of Software Engineering Process Innovations: The Case of Object-Orientation", invited presentation, *UCLA Anderson Graduate School of Management Information Systems Colloquium*, Los Angeles, California, October 1992.

"Adoption of Software Engineering Process Innovations: The Case of Object-Orientation", invited presentation, *Nanyang Technological University*, Singapore, November 1992.

"Software Development Measurement and Performance Evaluation", invited presentation, *Information Management Research Centre, Nanyang Technological University*, Singapore, November 1992.

"Recent Work in Agency Theory and Information Systems Management", invited presentation, *Nanyang Technological University*, Singapore, November 1992.

"Managing software delivery by measurement", one day seminar, *Key Centre for Strategic Management, University of Melbourne*, Melbourne, Australia, November 1992.

"Adoption of Software Engineering Process Innovations: The Case of Object-Orientation", invited presentation, *University of Melbourne Graduate School of Management*, Melbourne, Australia, November 1992.

"Software Development Measurement and Performance Evaluation", invited presentation, *Fujitsu Centre for Managing Information Technology in Organizations, Australian Graduate School of Management*, Sydney, Australia, November, 1992.

"Adoption of Software Engineering Process Innovations: The Case of Object-Orientation", invited presentation, *Faculty of Commerce, University of New South Wales*, Sydney, Australia, November 1992.

"Reliability of Function Points Measurement: A Field Experiment", invited presentation, *Wharton School Decision Sciences Department, University of Pennsylvania*, Philadelphia, Pennsylvania, November 1992.

"Software Development Measurement and Performance Evaluation", invited presentation, *Centre de recherche Informatique de Montreal*, Montreal, Canada, December 1992.

Session chair, "Economics of Software Contracting", *Fourth Workshop on Information Systems and Economics*, Dallas, Texas, December 1992.

Discussant, "A Comparison of Albrecht's Function Point and Symon's Mark II Metrics", *Thirteenth Annual International Conference on Information Systems*, Dallas, Texas, December 1992.

"Computerized Loan Origination Systems: A Case Study of the Electronic Markets Hypothesis", *26th Annual Hawaii International Conference on System Sciences*, Maui, Hawaii, January 1993.

"Adoption of Software Engineering Process Innovations: The Case of Object-Orientation", invited presentation, *Harvard Graduate School of Business Administration Seminar*, Boston, Massachusetts, April, 1993.

"Metrics for Object Oriented System Environments", invited presentation, *XVI Taller de Ingeneria de Sistemas*, Santiago, Chile, July 1993. (Simultaneous translation, Spanish-English, English-Spanish).

"Measurements of Software Delivery", plenary speaker, *XVI Taller de Ingeneria de Sistemas*, Santiago, Chile, July 1993. (Simultaneous translation, Spanish-English, English-Spanish).

"Determinants of the Demand for Microcomputer Software: An Econometric Analysis of the Microcomputer Spreadsheet Market", invited presentation, *Copenhagen Business School*, Copenhagen, Denmark, September 1993.

"Integration of economic theory and information technology teaching", invited presentation, *Fagintegrationsdag, Copenhagen Business School*, Copenhagen, Denmark, September 1993.

"Determinants of the Demand for Microcomputer Software: An Econometric Analysis of the Microcomputer Spreadsheet Market", invited presentation, *Boston University MIS Seminar Series,* Boston, Massachusetts, September 1993.

"Determinants of the Demand for Microcomputer Software: An Econometric Analysis of the Microcomputer Spreadsheet Market", invited presentation, *Georgia State University CIS Colloquium,* Atlanta, Georgia, September 1993.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", invited presentation, *University of California-Center for Research on Information Technology and Organizations Seminar,* Irvine, California, October 1993.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", invited presentation, *University of Southern California School of Business Administration Seminar,* Los Angeles, California, October 1993.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", invited presentation, *Stanford University Graduate School of Business Seminar,* Stanford, California, October 1993.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", invited presentation, *Carnegie Mellon University Graduate School of Industrial Administration Seminar,* Pittsburgh, Pennsylvania, October 1993.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", invited presentation, *Naval Post-Graduate School Seminar,* Monterey, California, October 1993.

"MOOSE: Metrics for Object-Oriented Systems Environments", *4th International Conference on Applications of Software Measurement*, Orlando, Florida, November 1993.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", *5th Workshop on Information Systems and Economics*, Orlando, Florida, December 1993.

"Determinants of the Demand for Microcomputer Software: An Econometric Analysis of the Microcomputer Spreadsheet Market", *14th International Conference on Information Systems,* Orlando, Florida, December 1993.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", invited presentation, *Wharton School Operations and Information Management Department*, *University of Pennsylvania*, Philadelphia, Pennsylvania, March, 1994.

"Remembering the past, embracing the future", featured guest speaker, *Pine-Richland High School* Commencement Exercises, June 1994.

"Metrics for Object-Oriented Software Engineering: A Managerial Perspective", invited presentation, *University of Rochester Simon School*, Rochester, New York, November 1994.

"Metrics for Object-Oriented Software Engineering: A Managerial Perspective", invited presentation, *Software Engineering Institute, Carnegie Mellon University*, Pittsburgh, Pennsylvania, December 1994.

"Metrics for Object-Oriented Software Engineering: A Managerial Perspective", invited presentation, *University of Southern California*, Los Angeles, California, January 1995.

"Metrics for Object-Oriented Software Engineering: A Managerial Perspective", invited presentation, *Naval Post-graduate School*, Monterey, California, January 1995.

"Metrics for Object-Oriented Software Engineering: A Managerial Perspective", invited presentation, *Georgia State University*, Atlanta, Georgia, February 1995.

"Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market", invited presentation, *University of Pittsburgh*, Pittsburgh, Pennsylvania, February 1995.

"Improving Software Engineering Management through Measurement" invited tutorial, *IEEE International Engineering Management Conference*, Singapore, June 1995.

"Competing Through Information Technology Standards: Surviving the Standards Jungle", keynote presentation, *IEEE International Engineering Management Conference*, Singapore, June 1995.

"Measuring the Unmeasurable", keynote presentation, *Bournemouth Metrics Workshop,* Bournemouth, United Kingdom, April 1996.

"The Illusory Diffusion of Software Process Innovations: An Examination of Assimilation Gaps", invited presentation, *City University of Hong Kong*, Kowloon, Hong Kong, April 1996.

"Object-Oriented Systems: Who's Adopting, Why, and How", invited presentation, *City University of Hong Kong*, Kowloon, Hong Kong, April 1996.

"Why is Software so Hard?", inaugural David M. Roderick lecture, *University of Pittsburgh*, Pittsburgh, PA, May 1996.

"Managing Emerging Technologies: The Case of Object Orientation", invited seminar, *University of Virginia Center for the Management of Information Technology*, Charlottesville, VA, July 1996.

"Empirical Studies of Software Maintenance: A Research Program" invited presentation, *UCLA Anderson Graduate School of Management Information Systems Colloquium*, Los Angeles, California, April 1997.

"Software Reuse: Theory and Practice", invited presentation, *MIT Center for Information Systems Research*, Cambridge, MA, June 1997.

"Adoption of New Software Process Technologies", invited presentation, *Software Engineering Institute SPIN Lecture*, March 1998.

"Adoption of New Software Process Technologies", invited presentation, *University of Dayton Distinguished Speaker Series*, April 1998.

"A Longitudinal Empirical Analysis of Software Evolution", *International Workshop on Principles of Software Evolution*, Kyoto, Japan, April 1998.

"Software Measurement in Action", invited presentation, *MIT Center for Information Systems Research*, Cambridge, MA, June 1998.

"Adoption of New Software Process Technologies", invited presentation, *Pennsylvania State University*, September 1998.

"Adoption of New Software Process Technologies: A Research Program", invited presentation, *Case Western Reserve University*, November 1998.

"Sobering Up Software Engineering Research", keynote lecture, *Third Annual Empirical Applications of Software Engineering*, Staffordshire, England, UK, April 1999.

"Managing Software Project Risk", *University of Melbourne Business School*, invited presentation, Melbourne, Australia, June 1999.

"Successful Adoption of New Software Process Technologies", invited presentation, *University of Melbourne Business School*, Melbourne, Australia, June 1999.

"Adoption of New Software Process Technologies: A Research Program", invited presentation, *Faculty of Commerce, University of New South Wales*, Sydney, Australia, June 1999.

"Metrics for Object Oriented Software: A retrospective", invited presentation, *Sixth International Symposium on Software Metrics*, Boca Raton, FL, November 1999.

"Programs in Electronic Commerce", panel chair, *Eleventh Workshop on Information Systems and Economics*, Charlotte, NC, December 1999.

"Adoption of Object-Oriented Technologies: A Research Program", invited presentation, *National Research Council of Canada Seminar on Object-Oriented Software Quality*, Ottawa, Canada, February 2000.

"Adoption of New Software Process Technologies: A Research Program", invited presentation, *University of Washington*, Seattle, WA, March 2000.

"Sobering Up Empirical Software Engineering Research", The 2000 Paul Rook Memorial Lecture keynote speech, *European Software Control and Metrics Conference*, Munich, Germany, April 2000.

"Modern Software Project Management", academic director and primary faculty, *Carnegie Mellon University Executive Education*, Pittsburgh, PA, May 2000. (Simultaneous translation, English-Japanese, Japanese-English).

"Managing Software Development", faculty, *Carnegie Mellon University Software Engineering Evangelists for Korea*, Pittsburgh, PA, June 2000.

"Adoption of Software Process Innovations: Lessons Learned from a Program of Research on Object-Oriented Technologies", invited presentation, *University of Indiana Kelley School of Business*, Indianapolis, Indiana, November 2000.

"Adoption of Software Process Innovations: Lessons Learned from a Program of Research on Object-Oriented Technologies", invited presentation, *University of Texas McCombs School of Business*, Austin, Texas, December 2000.

"Software Quality Management", academic director and faculty, *Carnegie Mellon University Executive Education*, Pittsburgh, PA, May 2001. (Simultaneous translation, English-Japanese, Japanese-English).

"How Development Decisions Affect Product Lifecycle Volatility: A Longitudinal Study of Software Change Histories", invited presentation, *Purdue University,* West Lafayette, Indiana, April 2003.

"How Development Decisions Affect Product Lifecycle Volatility: A Longitudinal Study of Software Change Histories", invited presentation, *Arizona State University,* Tempe, Arizona, May 2003.

"How Development Decisions Affect Product Lifecycle Volatility: A Longitudinal Study of Software Change Histories", invited presentation, *University of Calgary,* Calgary, Canada, May 2003.

"How Development Decisions Affect Product Lifecycle Volatility: A Longitudinal Study of Software Change Histories", invited presentation, *Texas A&M University,* College Station, Texas, October 2003.

"Environmental Volatility, Development Decisions and Software Volatility: A Longitudinal Analysis", invited presentation, *Ohio State University,* Columbus, Ohio, October 2004.

"Environmental Volatility, Development Decisions and Software Volatility: A Longitudinal Analysis", invited presentation, *University of California,* Irvine, California, November 2004.

"Environmental Volatility, Development Decisions and Software Volatility: A Longitudinal Analysis", invited presentation, *University of Southern California,* Los Angeles, California, November 2004.

"Publishing HCI Research in MIS Journals (panel)", invited presentation, *Third Annual Workshop on HCI Research in MIS*, Washington, DC, December 2004.

"An Academic In The Courtroom: Lessons Learned from the Microsoft Anti-Trust Trials", invited presentation, *Pittsburgh Executive Series for Business Administration*, Pittsburgh, PA, March 2005.

"Getting Your Research Published (panel)", invited presentation, *International Conference on Information Systems Doctoral Consortium*, Las Vegas, NV, December 2005.

"Program and Project Management", invited presentation, *Allegheny Conference Executive Education Program*, Pittsburgh, PA, February 2006.

"Project Management", discussion leader, *GaliLead Program, Carnegie Mellon University,* Pittsburgh, PA, June 2007.

"Standards Competition in the Presence of Digital Conversion Technology: An Empirical Analysis of the Flash Memory Card Market." *NET Institute Conference on Network Economics*, New York, NY, April 2008.

"Winners Take Some: The Impact of Conversion Technologies on Network Effects in Digital Goods Markets", invited presentation, *Georgia Institute of Technology's IT Management Distinguished Speaker Series*, Atlanta, GA, April 2008.

"Project Risk Management", invited presentation, *Carnegie Mellon University Tepper School Cognizant RMC2-AM Track Program*, Pittsburgh, PA, April 2009.

"High Impact IS Research", keynote speech, *International Conference on Information Systems (ICIS) Doctoral Consortium*, Fudan University, Shanghai, China December 2011

"Measuring the Value of Free Goods and Services on the Internet", discussant, *Workshop on Information Systems and Economics*, Shanghai, China December 2011.

"Structural Complexity and Programmer Team Strategy: An Experimental Test", invited presentation, *King Abdulaziz University, Jeddah, Saudi Arabia*, September 2012.

"Text, Ties, and Videocassettes: The Coming End of the Winner-Takes-All Era", keynote speech, *Americas Conference on Information Systems (AMCIS) WEB 2013*, Chicago, IL, August 2013.

"Software Maintenance as Technical Debt", invited presentation, *Katz Research Seminar Series*, Pittsburgh, PA, January 2015.

"Case Method Teaching: Prescription for Participant-Centered Learning", invited presentation, *University of Pittsburgh School of Pharmacy*, Pittsburgh, PA, April 2015.

"Technical Debt and the Reliability of Enterprise Software Systems", invited presentation, *Georgia Tech Sandy A. Slaughter Software Conference*, Atlanta, GA, May 2015.

"Research in Software Panel: Reflections and the Road Ahead", invited panelist, *Georgia Tech Sandy A. Slaughter Software Conference*, Atlanta, GA, May 2015.

University of Pittsburgh (Katz), Pittsburgh, Pennsylvania

Management of Information Systems Practicum [BMIS 2056]

Data Management I [BMIS 2523]

Data Management II [BMIS 2524]

Managing Software Design I [BMIS 2573]

Managing Software Design II [BMIS 2574]

Managing Software Design [BMIS 2584]

Information Systems: IT and Business Value [BMIS 2801]

Advanced Topics in MIS [BMIS 3015]

Economics and Information Systems [BMIS 3022]

Technology Innovation, Adoption, and Diffusion (PhD Seminar) [BMIS 3025]

Technology Innovation, Adoption, and Diffusion (MBA course) [BMIS 2679]

Information Systems [BMIS 2411, BACC 2411]

Technology Innovation and Adoption (EMBA elective) [BMIS 2685]

Teaching Awards:

Executive MBA Program Distinguished Professor of the Year Award (2015)

Executive MBA Program Distinguished Professor of the Year Award (2014)

Executive MBA Program Distinguished Professor of the Year Award (2007)

Executive MBA Program Distinguished Professor of the Year Award Runner-up (2003)

Katz Excellence in Teaching Award 2001-2002, 2002-2003, 2004-2005, 2005-2006, 2008-2009, 2009-2010, 2010-2011, 2011-2012, 2013-2014, 2014-2015.

MIT Sloan School of Management, Cambridge, Massachusetts

Decision Support Systems I [15.560]

Management of Information Systems [15.568]

Advanced Decision Support Systems [15.569]

Software Engineering Management [15.577]

Special Seminar in Information Technology: Distance Learning [15.579]

Workshop in Information Technology [15.599]

Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania

Management Information Systems [OPIM 210]

Software Engineering Management [OPIM 669] / Advanced Programming Methods [OPIM 319]

Carnegie Mellon University, Pittsburgh, Pennsylvania

School of Computer Science

Managing Software Development (co-taught) [15-673, 17-653]

Technology Innovation, Adoption, and Diffusion [17-645]

Graduate School of Industrial Administration

Systems Analysis and Design (TA) [45-878]

Cost Estimation and Analysis (TA) [45-804]

Production I [70-371]

LaRoche College, Pittsburgh, Pennsylvania

Data Processing and Data Management [HRM 400]

Editor:

Guest editor, *Decision Support Systems*, IS and Economics Special Issue 1990.

Editorial Board, *Journal of Organizational Computing and Electronic Commerce*, 1990-2006.

Associate Editor, *Communications of the ACM*, 1991-1995.

Editorial Board, *Journal of Software Quality*, 1991-present.

Editorial Board, *Information Systems Research*, 1991-1995.

Associate Editor, *MIS Quarterly*, 1991-1997.

Editorial Board, *Annals of Software Engineering*, 1994-present.

Editorial Board, *Empirical Software Engineering*, 1995-2002.

Departmental Editor, *Management Science*, 1997-2002.

Associate Editor, *IEEE Transactions on Software Engineering*, 1999-2002.

Editor-in-Chief, *Information Systems Research*, 2002-2004.

Senior Editor, *MIS Quarterly*, 2008-2011.

International Conference on Information Systems (ICIS) service

Program committee member,

Tenth International Conference on Information Systems (ICIS), Boston, MA, 1989.

Eleventh International Conference on Information Systems (ICIS), Copenhagen, Denmark, 1990.

Fifteenth International Conference on Information Systems (ICIS), Vancouver, Canada, 1994.

Executive committee member,

International Conference on Information Systems (ICIS), 1994-96.

Program co-chair,

Sixteenth International Conference on Information Systems (ICIS), Amsterdam, Netherlands, 1995.

Doctoral Consortium Faculty

Seventeenth International Conference on Information Systems (ICIS), Cleveland, Ohio, 1996.

Twenty-Fourth International Conference on Information Systems (ICIS), Seattle, Washington, 2003.

Twenty-Sixth International Conference on Information Systems (ICIS), Las Vegas, Nevada, 2005.

Keynoter, Thirty-Second International Conference on Information Systems (ICIS), Shanghai, China, 2011.

Best Doctoral dissertation committee member,

Eighteenth International Conference on Information Systems (ICIS), Atlanta, Georgia 1997.

Twenty-fifth International Conference on Information Systems (ICIS), Washington, DC, 2004.

Twenty-seventh International Conference on Information Systems (ICIS), Milwaukee, Wisconsin, 2006.

Best Paper prize committee member,
Twentieth International Conference on Information Systems (ICIS), Charlotte, NC, 1999.

Research paper track co-chair/associate editor,
Twenty-First International Conference on Information Systems (ICIS), Brisbane, Australia, 2000.

Twenty-Seventh International Conference on Information Systems (ICIS), Milwaukee, Wisconsin, 2006.

Thirtieth International Conference on Information Systems (ICIS), Phoenix, Arizona, 2009.

Other Professional Affiliations (chronological)
Founder and co-chair,
First Workshop on Information Systems and Economics (WISE), Cambridge, MA, 1989
Planning committee member and advisory board,
Workshop on Information Systems and Economics (WISE), 1990-present.

Advisory council member,
Third International Workshop on Computer-Aided Software Engineering, London, UK, 1989.

Fourth International Workshop on Computer-Aided Software Engineering, Irvine, CA, 1990.

Fifth International Workshop on Computer-Aided Software Engineering, Montreal, Canada, 1992.

Sixth International Workshop on Computer-Aided Software Engineering, Singapore, 1993.

Conference committee member,
Fourth Annual Oregon Workshop on Software Metrics, Silver Falls, OR, 1992.

Program committee member,
Second International Software Metrics Symposium, London, UK, 1994.

International Advisory Board member
IEEE Engineering Management Conference, Singapore, 1995

Associate,

National Software Council Planning Task Force

Member, Balloting group,

IEEE Standard for Software Productivity Metrics - P1045, Software Engineering Standards Subcommittee

IEEE Standard for Software Life Cycle Processes - P1448, Software Engineering Standards Subcommittee

External Program Review Committee,

University of Minnesota Graduate School Ph.D. program, 1995-96.

Program committee member,

International Workshop on Empirical Studies of Software Maintenance, Monterey, CA, 1996.

Program committee member,

IEEE 5th International Symposium on the Assessment of Software Tools and Technologies (SAST97), Pittsburgh, PA, 1997

External Examiner,

University of Western Ontario (Canada) Ph.D. thesis, 1997.

Member,

Beta Gamma Sigma (business school honor society).

Nominee,

University of Pittsburgh Chancellor's Distinguished Research Award, 1999-2000

University of Pittsburgh Chancellor's Distinguished Research Award, 2012-2013

Judge,

Second Carnegie Mellon University Master of Science in Electronic Commerce Practicum Project Contest (MSEC), Pittsburgh, PA, 2001.

Third Carnegie Mellon University Master of Science in Electronic Commerce Practicum Project Contest (MSEC), Pittsburgh, PA, 2002.

Marquis Who's Who in America,
58th Edition (2003), 59th Ed. (2004), 60th Ed. (2005), 61st Ed. (2006), 67th Ed. (2013).

Advisory Board Member,
CIO Institute, Carnegie Mellon University.

Senior Scholar Consortium,
International Conference on Information Systems (ICIS), Washington, DC, 2004.

ISI Highly Cited Researcher (top computer scientists worldwide)
Thomson Reuters, 2005-present

Program committee member,
Twenty-Seventh International Conference on Software Engineering (ICSE), St. Louis, MO, 2005.

Marquis Who's Who in Science and Engineering,
8th Edition (2004), 9th Edition (2006), 10th Edition (2007-8).

Academy of Management OCIS Division Junior Faculty Consortium Senior Faculty Leader,
Academy of Management, Honolulu, HI, 2005.

Marquis Who's Who in American Education,
7th Edition (2005), 8th Ed. (2007).

Advisory Board,
Social Science Research Network

Affiliate,
Sloan Industry Studies Program 2007.

Counselor,
Junior Faculty Consortium, Americas Conference on Information Systems, Toronto, Canada, 2008.

Program Committee,
4th Software Engineering Conference (Russia), Moscow, Russia, 2008.

Program Committee,

Software Development Governance workshop, 30th International Conference on Software Engineering (ICSE), Leipzig, Germany, 2008.

Software Development Governance workshop, 31st International Conference on Software Engineering (ICSE), Vancouver, Canada, 2009.

Member,

Senior Scholars Best Publications Committee, 2008-2009, 2009-2010, 2010-2011, 2011-2012, 2012-2013.

Honoree,

Katz Excellence in Research Award 2009-10, 2010-11, 2011-12, 2012-2013, 2014-2015.

International Affiliate,

King Abdulaziz University, Jeddah, Saudi Arabia, 2012-.

Area Head,

Katz Graduate School of Business Information Systems and Technology Management Area, 2013-present.

Distinguished Fellow,

INFORMS Information Systems Society, 2013.

Reviewer:

ACM Computing Surveys; *ACM Transactions on Software Engineering and Methodology, Accounting, Management, and Information Technologies*; Academy of Management Conference, Addison-Wesley Publishing, Australian Research Council; *Automated Software Engineering*; *Communications of the ACM*; Hawaii International Conference on Systems Sciences; Hong Kong Research Grants Council; IEEE-CS International Software Metrics Symposium; *IEEE Software Risk Management Planning Group; IEEE Transactions on Software Engineering*; International Conference on Software Engineering; *Information and Software Technology; Information Systems Research*; International Conference on Information Systems; *Journal of the Association for Information Systems*, *Journal of Industrial Economics*; *Journal of Organizational Computing*; *Journal of Software Maintenance and Evolution*, *Journal of Systems and Software*; *Management Science*; McGraw-Hill, *MIS Quarterly*; *Organization Science*; Pearson Education, Prentice-Hall; Prentice-Hall Australia; Reviewer, Science Foundation Ireland, *Science of Computer Programming*, *Sloan Management Review*; *Software Engineering Journal*; UCL Press (UK); Wadsworth Publishing.

APPENDIX C – Materials Considered

A. Public Sources

- Aaron Williamson, *Lawsuit threatens to break new ground on the GPL and software licensing issues,* Opensource (Jul 30, 2014), https://opensource.com/law/14/7/lawsuit-threatens-break-new-ground-gpl-and-software-licensing-issues.
- ABSTRACT WINDOW TOOLKIT (AWT), ORACLE, http://docs.oracle.com/javase/7/docs/technotes/guides/awt/index.html (last visited January 6, 2016).
- ADD-ONS, MOZILLA, https://addons.mozilla.org/en-US/firefox/addon/apk-downloader/ (last visited January 6, 2016).
- ANDROID FRAMEWORK CLASSES AND SERVICES, GOOGLE GIT, https://android.googlesource.com/platform/frameworks/base (last visited January 6, 2016).
- ANDROID- AN OPEN HANDSET ALLIANCE PROJECT, ANDROID, https://web.archive.org/web/20090124043917/http://code.google.com/android/kb/licensingandoss.html#apache2 (last visited January 6, 2016).
- ANDROID Source Code Repositories
- ANNOTATIONS, ORACLE, http://docs.oracle.com/javase/7/docs/technotes/guides/language/annotations.html (last visited January 6, 2016).
- ANNOUNCING ANDROID 10 SDK RELEASE, ANDROID, http://android-developers.blogspot.com/2008/09/announcing-android-10-sdk-release-1.html (last visited January 8, 2016).
- Billy Ehrenberg, *How Much is Your Personal Data Worth*, The Guardian (Apr 8, 2014), http://www.theguardian.com/news/datablog/2014/apr/22/how-much-is-personal-data-worth.
- Brian Rubin, *Google's ARC Welder Gives You a Glimpse of an Android-Anywhere Future,* readwrite (April 2, 2015), http://readwrite.com/2015/04/02/arc-welder-android-apps-to-chrome.
- BRILLO, GOOGLE, https://developers.google.com/brillo/?hl=en (last visited January 8, 2016).
- CFR- ANOTHER JAVA DECOMPILER, http://www.benf.org/other/cfr/ (last visited January 6, 2016).
- Christine Erickson*, Google Privacy: Five Things the Tech Giant Does with Your Data,* Mashable (Mar 1, 2012), http://mashable.com/2012/03/01/google-privacy-data-policy/#0nnyMEm_bZqH.
- CLASS CHARSET, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/nio/charset/Charset.html (last visited January 6, 2016).
- CLASS FONT, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/awt/Font.html (last visited January 6, 2016).
- CLDC 1.1, ORACLE, http://docs.oracle.com/javame/config/cldc/ref-impl/cldc1.1/jsr139 (last visited January 6, 2016).
- CONNECTED DEVICE CONFIGURATION (CDC), VERSION 1.1.2, ORACLE, http://docs.oracle.com/javame/config/cdc/ref-impl/cdc1.1.2/jsr218 (last visited January 6, 2016).
- CONNECTED LIMITED DEVICE CONFIGURATION (CLDC); JSR 139, ORACLE, http://www.oracle.com/technetwork/java/cldc-141990.html (last visited Jan. 5, 2016).
- Dan Morrill, *Announcing the Android 1.0 SDK, release 1,* Android Developers Blog (Sep 23, 2008), http://android-developers.blogspot.com/2008/09/announcing-android-10-sdk-release-1.html.
- DEPENDENCY ANALYSIS, SCITOOLSUNDERSTAND, https://scitools.com/features/#feature-category-dependency-analysis (last visited January 6, 2016).
- DEX2JAR, SOURCEFORGE, http://sourceforge.net/projects/dex2jar/ (last visited January 6, 2016).
- Diego Puppin & Fabrizio Silvestri, *The Social Network of Java Classes,* Institute of Information Science and Technologies (Apr 2006), http://pomino.isti.cnr.it/~silvestr/wp-content/uploads/2011/02/sac2006.pdf.
- Eric Ravenscraft, *ARC Welder Lets Devs (And You) Test Android Apps in Chrome,* LifeHacker (April 1, 2015), http://lifehacker.com/arc-welder-lets-devs-and-you-test-android-apps-in-chr-1694971713.
- Erick Schonfeld, *Breaking: Google Announces Android and Open Handset Alliance,* TechCrunch (Nov 5, 2007), http://techcrunch.com/2007/11/05/breaking-google-announces-android-and-open-handset-alliance.
- Fritzinger, J. Steven, and Marianne Mueller. "Java security." White Paper, Sun Microsystems, Inc (1996), available at http://net.uom.gr/Books/Manuals/whitepaper.pdf (2003)
- GET STARTED WITH ARC, CHROME, https://developer.chrome.com/apps/getstarted_arc (last visited January 8, 2016).

- GNU GENERAL PUBLIC LICENSE, VERSION 2, GNU OPERATING SYSTEM, http://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html (last visited Jan 5, 2016).
- GOOGLE PLAY APPS CRAWLER, GITHUB, https://github.com/MarcelloLins/GooglePlayAppsCrawler (last visited January 6, 2016).
- GOOGLE PLAY, GOOGLE, https://play.google.com/store/apps (last visited January 6, 2016).
- HIGH PRODUCTIVITY SOFTWARE FOR COMPLEX NETWORKS, NETWORKX, http://networkx.github.io (last visited January 6, 2016).
- HOW TO USE GOOGLE'S ARC WELDER TO RUN ANDROID APPS IN CHROME, HOW TO GEEK, http://www.howtogeek.com/214734/how-to-use-googles-arc-welder-to-run-android-apps-in-chrome (last visited January 6, 2016).
- INSTALLING THE ANDROID SDK, ANDROID, http://developer.android.com/sdk/installing/index.html (last visited January 6, 2016).
- INTERFACE ACL, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/security/acl/Acl.html (last visited January 6, 2016).
- INTRODUCTION TO SOCIAL NETWORK METHODS, UC RIVERSIDE, http://faculty.ucr.edu/~hanneman/nettext/C10_Centrality.html (last visited January 6, 2016).
- J2ME CLDC API, ORACLE, https://docs.oracle.com/javame/config/cldc/ref-impl/cldc1.0/cldcapi.pdf (last visited January 6, 2016).
- J2SE 1.4.2, ORACLE, http://www.oracle.com/technetwork/java/javase/index-jsp-138567.html (last visited Jan 5, 2016).
- James O'Toole, *Mobile Apps Overtake PC Internet Usage in US*, CNN Money (Feb 28, 2014) http://money.cnn.com/2014/02/28/technology/mobile/mobile-apps-internet.
- Jason Compton, *Java Time Line*, Chicago Tribune (Sep 11, 1998), http://articles.chicagotribune.com/1998-09-11/news/9901120092_1_java-developer-conference-javaos-netscape.
- JAVA 2 PLATFORM STANDARD EDITION 5.0 API SPECIFICATION, ORACLE, https://docs.oracle.com/javase/1.5.0/docs/api/ (last visited January 6, 2016).
- JAVA 20 YEARS, ORACLE, http://oracle.com.edgesuite.net/timeline/java/ (last visited January 8, 2016).
- JAVA ME TECHNOLOGY – CDC, ORACLE, http://www.oracle.com/technetwork/java/javame/tech/index-jsp-139293.html (last visited Jan. 5, 2016).
- JAVA ME, CDC 1.0.2 IS DEFINED IN JSR-36, http://download.oracle.com/otn-pub/jcp/7825-j2me_cdc-1.0a-mr-spec-oth-JSpec/j2me_cdc-1_0a-mr-spec.zip (last visited January 6, 2016).
- JAVA ME, CDC 1.1.2 IS DEFINED IN JSR-218. http://download.oracle.com/otn-pub/jcp/cdc-1.1-fr-eval-oth-spec-JSpec/cdc-1_1-fr-spec.zip; see also
- JAVA ME, CLDC 1.0 IS DEFINED IN JSR-30. See http://download.oracle.com/otn-pub/jcp/7587-j2me_cldc-1.0a-fr-spec-oth-JSpec/cldc-1_0a-spec.zip
- JAVA PLATFORM, MICRO EDITION (JAVA ME), ORACLE, http://www.oracle.com/technetwork/java/embedded/javame/index.html (last visited Jan 5, 2016).
- JAVA PLATFORM, STANDARD EDITION 7 API SPECIFICATION, ORACLE, http://docs.oracle.com/javase/7/docs/api/index.html (last visited January 6, 2016).
- JAVA PROGRAMMING ENVIRONMENT AND THE JAVA RUNTIME ENVIRONMENT (JRE), ORACLE, https://docs.oracle.com/cd/E19455-01/806-3461/6jck06gqd/index.html (Last visited Jan. 05, 2016).
- JAVA REFLECTION API, ORACLE, http://docs.oracle.com/javase/7/docs/technotes/guides/reflection/index.html (last visited January 6, 2016).
- JAVA SE AT A GLANCE, ORACLE, http://www.oracle.com/technetwork/java/javase/overview/index.html (last visited Jan 5, 2016).
- JAVA SE VERSIONS HISTORY, CODEJAVA, http://www.codejava.net/java-se/java-se-versions-history (Last visited January 5, 2016).
- JAVA VIRTUAL MACHINE, BUSINESS DICTIONARY, http://www.businessdictionary.com/definition/Java-virtual-machine-JVM.html (last visited Jan 5, 2016).
- JAVA VIRTUAL MACHINE, BUSINESS DICTIONARY, http://www.businessdictionary.com/definition/Java-virtual-machine-JVM.html (last visited Jan 5, 2016).
- JAVA, http://mail.openjdk.java.net/pipermail/announce/2007-May.txt, (last visited January 8, 2016).
- JAVABEANS FAQ: GENERAL QUESTIONS, ORACLE, http://www.oracle.com/technetwork/java/javase/faq-135947.html (last visited January 6, 2016).
- JAVADOC TOOL, ORACLE, http://www.oracle.com/technetwork/java/javase/documentation/javadoc-137458.html (last visited January 6, 2016).

- JAVAONE DRAWS 10,000 ATTENDEES—BREAKS ATTENDANCE RECORDS, JAVA SPOT NEWS (April 8, 1997), https://web.archive.org/web/19990117032326/http://java.sun.com/pr/1997/april/spotnews/sn970408.html
- JAVATM PLATFORM STANDARD EDITION 7 NAMES AND VERSIONS, ORACLE, http://www.oracle.com/technetwork/java/javase/jdk7-naming-418744.html (last visited Jan. 5, 2016).
- JDK 5.0 DOCUMENTATION, JAVA, https://web.archive.org/web/20100330080522/http://java.sun.com/j2se/1.5.0/docs/index.html (last visited January 6, 2016).
- Jeremy Kirk, *VMware sued for alleged GPL license infractions,* PCWorld (Mar 5, 2015), http://www.pcworld.com/article/2893852/vmware-sued-for-alleged-gpl-license-infractions.html.
- John Saddington, *The Story of Busybox and the First GPL Lawsuit,* Torque Mag (Mar 15, 2013), http://torquemag.io/busybox.
- JSR OVERVIEW, JAVA COMMUNITY PROCESS, https://jcp.org/en/jsr/overview (last visited Jan. 5, 2016).
- Kate Knibbs, *Why Android Phones Now Come With So Many More Google Apps Than Before*, Gizmodo (Sep 26, 2014), http://gizmodo.com/why-android-phones-now-come-with-so-many-more-google-ap-1639529342.
- Kent German, *A Brief History of Android Phones*, CNET (August 2, 2011), http://www.cnet.com/news/a-brief-history-of-android-phones.
- Kieron Murphy, *The pros and cons of JDK1.1,* JavaWorld (Apr 1, 1997), http://www.javaworld.com/article/2077643/core-java/the-pros-and-cons-of-jdk-1-1.html.
- LEARN ABOUT JAVA TECHNOLOGY, JAVA, https://www.java.com/en/about (last visited January 6, 2016).
- LESSON: PACKING PROGRAMS IN JAR FILES, ORACLE, https://docs.oracle.com/javase/tutorial/deployment/jar/index.html (last visited January 6, 2016).
- Mariva H. Aviram, *What Sun won't tell you about JavaOne*, JavaWorld (August 20, 1999), http://www.javaworld.com/article/2076471/what-sun-won-t-tell-you-about-javaone.html.
- METHOD FOR NODE RANKING IN A LIKED DATA BASE, GOOGLE PATENT (SEP 4, 2001), http://www.google.com/patents/US6285999.
- MOBILE INFORMATION DEVICE PROFILE (MIDP); JSR 118, ORACLE, http://www.oracle.com/technetwork/java/index-jsp-138820.html (last visited Jan. 5, 2016).
- NETWORK SCIENCE, http://www.sci.unich.it/~francesc/teaching/network (last visited January 6, 2016).
- NO. WE DO NOT SELL YOUR PERSONAL INFORMATION, GOOGLE, https://privacy.google.com/about-ads.html (last visited January 1, 2016)
- OPENJDK, http://openjdk.java.net/ (Last visited Jan. 05, 2016).
- ORACLE JAVA ARCHIVE, ORACLE, http://www.oracle.com/technetwork/java/javase/archive-139210.html (last visited January 8, 2016).
- PACKAGE INDEX, ANDROID DEVELOPER BLOG, http://developer.android.com/reference/packages.html (last visited January 6, 2016).
- PACKAGE JAVA.BEANS, ORACLE, http://docs.oracle.com/javase/7/docs/api/java/beans/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.IO DESCRIPTION, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html#package_description (last visited January 6, 2016).
- PACKAGE JAVA.LANG, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/lang/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.LANG.REF, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/lang/ref/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.LANG.REFLECT, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/lang/reflect/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.NET, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/net/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.NIO.CHANNELS, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/nio/channels/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.NIO.CHANNELS.SPI, ORACLE, http://docs.oracle.com/javase/7/docs/api/java/nio/channels/spi/package-summary.html (last visited January 6, 2016).

- PACKAGE JAVA.NIO.CHARSET.SPI, ORACLE, http://docs.oracle.com/javase/7/docs/api/java/nio/charset/spi/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.SECURITY, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/security/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.SECURITY.CERT DESCRIPTION, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/security/cert/package-summary.html#package_description (last visited January 6, 2016).
- PACKAGE JAVA.SQL, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/sql/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.TEXT, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/text/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVA.UTIL, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/util/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVAX.CRYPTO, ORACLE, https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html (last visited January 6, 2016).
- PACKAGE JAVAX.SQL, ORACLE, http://docs.oracle.com/javase/7/docs/api/javax/sql/package-summary.html (last visited January 6, 2016).
- PAKCAGE JAVAX.NET, ORACLE, http://docs.oracle.com/javase/7/docs/api/javax/net/package-summary.html (last visited January 6, 2016).
- Paul Krill, *Open Source Java for Android? Don't Bet on it,* JavaWorld (Nov 20, 2012), http://www.javaworld.com/article/2078666/mobile-java/open-source-java-for-android--don-t-bet-on-it.html.
- PERSONAL BASIS PROFILE OVERVIEW, ORACLE, http://www.oracle.com/technetwork/java/overview-141255.html (last visited January 6, 2016).
- Reginald Sawilla & Xinming Ou, *Identifying Critical Attack Assets in Dependency Attack Graphs,* Kansas State University (Sep 2008), http://people.cis.ksu.edu/~xou/publications/drdc08.pdf.
- Ron Amadeo, *Android 5.0 Lollipop, Thoroughly Reviewed,* arstechnica (Nov 12, 2014), http://arstechnica.com/gadgets/2014/11/android-5-0-lollipop-thoroughly-reviewed.
- Rongcun Wang, Rubing Huang & Binbin Qu, *Network-Based Analysis of Software Change Propagation,* The Scientific World Journal (Oct 17, 2013), http://www.hindawi.com/journals/tswj/2014/237243.
- Ryan Paul, *Why Google Chose the Apache Software License over GPL v2 for Android,* arstechnica (Nov 6, 2007), http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2.
- SELENIUM HQ BROWSER AUTOMATION, SELENIUM HQ, http://www.seleniumhq.org/projects/ide/ (last visited January 6, 2016).
- SMARTPHONE OS WARS: DEVELOP FOR WHICH PLATFORMS? PART I, LOCALYTICS (Jun 18, 2009), https://www.localytics.com/smartphone-os-wars-develop-for-which-platforms-part-I.
- TCK PROJECT PLANNING AND DEVELOPMENT GUIDE, (August, 2003), at pg. 2 https://jcp.org/aboutJava/communityprocess/ec-public/TCK-docs/ppg.pdf.
- TCK PROJECT PLANNING AND DEVELOPMENT GUIDE, (August, 2003), at pg. 2 https://jcp.org/aboutJava/communityprocess/ec-public/TCK-docs/ppg.pdf.
- THE PAGERANK CITATION RANKING: BRINGING ORDER TO THE WEB, STANFORD (JAN 29, 1998), http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.
- THE WORLD'S MOST VALUABLE BRANDS, FORBES http://www.forbes.com/companies/google (last visited January 6, 2016).
- Tian, Y. et al, "What Are the Characteristics of High-Rated Apps? A Case Study on Free Android Applications", *31st IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2015.
- TIOBE INDEX FOR JANUARY 2016, TIOBE, http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html (last visited January 6, 2016).
- USER SPACE, WIKIPEDIA, https://en.wikipedia.org/wiki/User_space (last visited January 6, 2016).
- Version 1.0.2 was obtained from the Taiwan National Central University JDK Archives *available at* http://in.ncu.edu.tw/center5/java/jdk/JDK-1_0_2-win32-x86.exe.
- Versions 1.1.3 to 1.8.0 were downloaded from Oracle's JDK Archives *available at* http://www.oracle.com/technetwork/java/javase/archive-139210.html.
- WHAT IS ANDROID, ANDROID DEVELOPERS, https://web.archive.org/web/20090228170145/http://developer.android.com/guide/basics/what-is-android.html (last visited January 6, 2016).

- WHAT IS J2ME OR JAVA ME?, ORACLE, https://java.com/en/download/faq/whatis_j2me.xml (last visited Jan 5, 2016).
- XML TO CSV CONVERSION TOOL, CODEPLEX, https://xmltocsv.codeplex.com (last visited January 6, 2016).

B. Court Documents

- Amici Curiae Brief of Scott McNealy and Brian Sutphin, Feb. 22, 2013
- Astrachan Tr. 2212
- Schmidt, Tr. 1456:15-19, 1457:19-25, 1458:1-16
- Screven, Tr. 512:24-513:2
- TX 13, GOOGLE-01-00019511
- TX 134 GOOGLE-01-00018143
- TX 147, GOOGLE-01-00023889
- TX 158, GOOGLE-01-00025575-587
- TX 172, GOOGLE-01-00029329
- TX 173, GOOGLE-01-00029331
- TX 21, GOOGLE-02-00111218
- TX 23, GOOGLE-04-00055098
- TX 230, GOOGLE-02-00020474
- TX 24, GOOGLE-01-00025375
- TX 246, GOOGLE-02-00089698-99
- TX 2488
- TX 25 (Java Virtual Machine Specification, p. 356 of 488).
- TX 416, GOOGLE-38-00020572
- TX 498
- TX 610.1
- United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022.

C. Produced Documents

- GOOG-00022382
- GOOG-00273867
- GOOGLE-00-00001772
- GOOGLE-01-00019511-513
- GOOGLE-01-00019527
- GOOGLE-22-00280869-GOOGLE-22-00280977
- OAGOOGL0004381807
- OAGOOGLE0000191668
- OAGOOGLE0000609523
- OAGOOGLE0001817230
- OAGOOGLE0003997531-OAGOOGLE0003997532
- OAGOOGLE0007356223
- OAGOOGLE0100072596-98
- OAGOOGLE0100405426-28
- OAGOOGLE0100628653

D. Depositions

- Deposition of Daniel Morrill (July 12, 2011)
- Deposition of Daniel Bornstein (July 22, 2011)
- Deposition of Vineet Gupta (July 26, 2011)
- Deposition of Owen Astrachan (April 27, 2012)
- Deposition of Anwar Ghuloum (December 9, 2015)

APPENDIX D – List of "Top" Apps

Name	App Name	Installations
Facebook	com.facebook.katana	1,000,000,000 - 5,000,000,000
Messenger	com.facebook.orca	1,000,000,000 - 5,000,000,000
Google Play Books	com.google.android.apps.books	1,000,000,000 - 5,000,000,000
Google Drive	com.google.android.apps.docs	1,000,000,000 - 5,000,000,000
Google Play Newsstand	com.google.android.apps.magazines	1,000,000,000 - 5,000,000,000
Maps	com.google.android.apps.maps	1,000,000,000 - 5,000,000,000
Gmail	com.google.android.gm	1,000,000,000 - 5,000,000,000
Google	com.google.android.googlequicksearch	1,000,000,000 - 5,000,000,000
Google Talkback	com.google.android.marvin.talkback	1,000,000,000 - 5,000,000,000
Google Play Music	com.google.android.music	1,000,000,000 - 5,000,000,000
Google Play Games	com.google.android.play.games	1,000,000,000 - 5,000,000,000
Google Text-to-speech	com.google.android.tts	1,000,000,000 - 5,000,000,000
Google Play Movies & TV	com.google.android.videos	1,000,000,000 - 5,000,000,000
WhatsApp Messenger	com.whatsapp	1,000,000,000 - 5,000,000,000
Clean Master (Boost & AppLock)	com.cleanmaster.security	500,000,000 - 1,000,000,000
Google Street View	com.google.android.street	500,000,000 - 1,000,000,000
Instagram	com.instagram.android	500,000,000 - 1,000,000,000
Samsung Push Service	com.sec.spp.push	500,000,000 - 1,000,000,000
Skype - free IM & video calls	com.skype.raider	500,000,000 - 1,000,000,000
APUS Launcher-Small,Fast,Boost	com.apusapps.launcher	100,000,000 - 500,000,000
Mobile Security & Antivirus	com.avast.android.mobilesecurity	100,000,000 - 500,000,000
BBM	com.bbm	100,000,000 - 500,000,000
Ant Smasher, Best Free Game	com.bestcoolfungames.antsmasher	100,000,000 - 500,000,000
Blurb Checkout	com.blurb.checkout	100,000,000 - 500,000,000

CM Security Antivirus AppLock	com.cleanmaster.mguard	100,000,000 - 500,000,000
Cymera - Photo Editor, Collage	com.cyworld.camera	100,000,000 - 500,000,000
Tiny Flashlight + LED	com.devuni.flashlight	100,000,000 - 500,000,000
DU Speed Booster Cache Cleaner	com.dianxinos.optimizer.duplay	100,000,000 - 500,000,000
AppLock	com.domobile.applock	100,000,000 - 500,000,000
Dropbox	com.dropbox.android	100,000,000 - 500,000,000
ANT+ Plugins Service	com.dsi.ant.plugins.antplus	100,000,000 - 500,000,000
ANT Radio Service	com.dsi.ant.service.socket	100,000,000 - 500,000,000
Real Racing 3	com.ea.games.r3_na	100,000,000 - 500,000,000
eBay	com.ebay.mobile	100,000,000 - 500,000,000
ES File Explorer File Manager	com.estrongs.android.pop	100,000,000 - 500,000,000
Evernote	com.evernote	100,000,000 - 500,000,000
Hill Climb Racing	com.fingersoft.hillclimb	100,000,000 - 500,000,000
Pool Billiards Pro	com.forthblue.pool	100,000,000 - 500,000,000
Despicable Me	com.gameloft.android.ANMP.GloftDMHM	100,000,000 - 500,000,000
GO Launcher -Theme & Wallpaper	com.gau.go.launcherex	100,000,000 - 500,000,000
Cloud Print	com.google.android.apps.cloudprint	100,000,000 - 500,000,000
Google News & Weather	com.google.android.apps.genie.geniewidget	100,000,000 - 500,000,000
Google Translate	com.google.android.apps.translate	100,000,000 - 500,000,000
Google Calendar	com.google.android.calendar	100,000,000 - 500,000,000
Google Keyboard	com.google.android.inputmethod.latin	100,000,000 - 500,000,000
Google Earth	com.google.earth	100,000,000 - 500,000,000
Fruit Ninja Free	com.halfbrick.fruitninjafree	100,000,000 - 500,000,000
Jetpack Joyride	com.halfbrick.jetpackjoyride	100,000,000 - 500,000,000
HP Print Service Plugin	com.hp.android.printservice	100,000,000 - 500,000,000
Battery Doctor (Battery Saver)	com.ijinshan.kbatterydoctor_en	100,000,000 - 500,000,000
Temple Run 2	com.imangi.templerun2	100,000,000 - 500,000,000
imo free video calls and chat	com.imo.android.imoim	100,000,000 - 500,000,000
GO SMS Pro	com.jb.gosms	100,000,000 - 500,000,000
Candy Crush Saga	com.king.candycrushsaga	100,000,000 - 500,000,000
Candy Crush Soda Saga	com.king.candycrushsodasaga	100,000,000 - 500,000,000
Farm Heroes Saga	com.king.farmheroessaga	100,000,000 - 500,000,000
Pet Rescue Saga	com.king.petrescuesaga	100,000,000 - 500,000,000
Omni Swipe	com.lazyswipe	100,000,000 - 500,000,000
Lookout Security & Antivirus	com.lookout	100,000,000 - 500,000,000
8 Ball Pool	com.miniclip.eightballpool	100,000,000 - 500,000,000
MX Player	com.mxtech.videoplayer.ad	100,000,000 - 500,000,000
Glow Hockey	com.natenai.glowhockey	100,000,000 - 500,000,000
Netflix	com.netflix.mediaclient	100,000,000 - 500,000,000
My Talking Angela	com.outfit7.mytalkingangelafree	100,000,000 - 500,000,000

My Talking Tom	com.outfit7.mytalkingtomfree	100,000,000 - 500,000,000
Talking Ginger	com.outfit7.talkinggingerfree	100,000,000 - 500,000,000
Talking Tom Cat 2	com.outfit7.talkingtom2free	100,000,000 - 500,000,000
PandoraÂ® Radio	com.pandora.android	100,000,000 - 500,000,000
PicsArt Photo Studio	com.picsart.studio	100,000,000 - 500,000,000
360 Security - Antivirus Boost	com.qihoo.security	100,000,000 - 500,000,000
Photo Grid - Collage Maker	com.roidapp.photogrid	100,000,000 - 500,000,000
Angry Birds Rio	com.rovio.angrybirdsrio	100,000,000 - 500,000,000
Angry Birds Seasons	com.rovio.angrybirdsseasons	100,000,000 - 500,000,000
S Health -Fitness Diet Tracker	com.sec.android.app.shealth	100,000,000 - 500,000,000
Samsung Print Service Plugin	com.sec.app.samsungprintservice	100,000,000 - 500,000,000
Tango - Free Video Call & Chat	com.sgiggle.production	100,000,000 - 500,000,000
Shazam	com.shazam.android	100,000,000 - 500,000,000
Snapchat	com.snapchat.android	100,000,000 - 500,000,000
Smart Connect	com.sonyericsson.extras.liveware	100,000,000 - 500,000,000
Spotify Music	com.spotify.music	100,000,000 - 500,000,000
Clash of Clans	com.supercell.clashofclans	100,000,000 - 500,000,000
Hay Day	com.supercell.hayday	100,000,000 - 500,000,000
4shared	com.sync.note	100,000,000 - 500,000,000
TripAdvisor Hotels Flights	com.tripadvisor.tripadvisor	100,000,000 - 500,000,000
Twitter	com.twitter.android	100,000,000 - 500,000,000
UC Browser - Fast Download	com.UCMobile.intl	100,000,000 - 500,000,000
Don't Tap The White Tile	com.umonistudio.tile	100,000,000 - 500,000,000
Retrica	com.venticake.retrica	100,000,000 - 500,000,000
Viber	com.viber.voip	100,000,000 - 500,000,000
Yahoo Mail â€“ Free Email App	com.yahoo.mobile.client.android.mail	100,000,000 - 500,000,000
Cut the Rope FULL FREE	com.zeptolab.ctr.ads	100,000,000 - 500,000,000
Flipboard: Your News Magazine	flipboard.apps	100,000,000 - 500,000,000
Kik	kik.android	100,000,000 - 500,000,000
Pou	me.pou.app	100,000,000 - 500,000,000
ZEDGEâ„¢ Ringtones & Wallpapers	net.zedge.android	100,000,000 - 500,000,000
Firefox Browser for Android	org.mozilla.firefox	100,000,000 - 500,000,000
TuneIn Radio - Radio & Music	tunein.player	100,000,000 - 500,000,000
Peel Smart Remote (WatchONâ„¢)	tv.peel.app	100,000,000 - 500,000,000
Camera360 Ultimate	vStudio.Android.Camera360	100,000,000 - 500,000,000

APPENDIX E – Excluded Apps

Name	App_ID	Installations	Reason For Exclusion
Adobe AIR	com.adobe.air	100,000,000 - 500,000,000	>25% Unknown App Dependencies
Chrome Browser - Google	com.android.chrome	1,000,000,000 - 5,000,000,000	>25% Unknown App Dependencies
Hungry Shark Evolution	com.fgol.HungrySharkEvolution	100,000,000 - 500,000,000	>25% Unknown App Dependencies
Subway Surfers	com.kiloo.subwaysurf	500,000,000 - 1,000,000,000	>25% Unknown App Dependencies
Angry Birds	com.rovio.angrybirds	100,000,000 - 500,000,000	>25% Unknown App Dependencies
Shoot Bubble Deluxe	com.shootbubble.bubbledexlue	100,000,000 - 500,000,000	>25% Unknown App Dependencies
Traffic Racer	com.skgames.trafficracer	100,000,000 - 500,000,000	>25% Unknown App Dependencies
Super-Bright LED Flashlight	com.surpax.ledflashlight.panel.apk	100,000,000 - 500,000,000	>25% Unknown App Dependencies
3D Bowling	com.threed.bowling	100,000,000 - 500,000,000	>25% Unknown App Dependencies
DU Battery Saver(power saver)	com.dianxinos.dxbs	100,000,000 - 500,000,000	>25% Unknown App Dependencies
YouTube	com.google.android.youtube	1,000,000,000 - 5,000,000,000	>25% Unknown App Dependencies
Hola Launcher	com.hola.launcher	100,000,000 - 500,000,000	>25% Unknown App Dependencies
KakaoTalk: Free Calls & Text	com.kakao.talk	100,000,000 - 500,000,000	>25% Unknown App Dependencies
WeChat	com.tencent.mm	100,000,000 - 500,000,000	>25% Unknown App Dependencies
Drag Racing	com.creativemobile.DragRacing	100,000,000 - 500,000,000	>25% Unknown App Dependencies
LINE: Free Calls & Messages	jp.naver.line.android	100,000,000 - 500,000,000	>25% Unknown App Dependencies
Hangouts	com.google.android.talk	1,000,000,000 - 5,000,000,000	>25% Unknown App Dependencies
Adobe Acrobat Reader	com.adobe.reader	100,000,000 - 500,000,000	Unable to detect app package
Amazon Kindle	com.amazon.kindle	100,000,000 - 500,000,000	Unable to detect app package
Barcode Scanner	com.gamma.scan	100,000,000 - 500,000,000	Unable to detect app package
Google Photos	com.google.android.apps.photos	100,000,000 - 500,000,000	Unable to detect app package
Temple Run	com.imangi.templerun	100,000,000 - 500,000,000	Unable to detect app package
Talking Tom Cat	com.outfit7.talkingtom	100,000,000 - 500,000,000	Unable to detect app package
Google+		1,000,000,000 -	Incompatible with

		5,000,000,000	Downloading Device
Android System WebView		100,000,000 - 500,000,000	Incompatible with Downloading Device
WhatsApp Wallpaper		100,000,000 - 500,000,000	Incompatible with Downloading Device
Samsung Security Policy Update		100,000,000 - 500,000,000	Incompatible with Downloading Device
Beaming Service for Samsung		100,000,000 - 500,000,000	Incompatible with Downloading Device

APPENDIX F – PHP Parser Script for Javadoc HTML Documentation

```
<pre>

<?php



require('simpledomparser.php');

function text_similarity($string_a, $string_b){

        $similar_text = '';

        similar_text($string_a, $string_b, $percent);

        $similar_text_a = (100.0 - $percent) * 1.0 * strlen($string_a.$string_b);

        similar_text($string_b, $string_a, $percent);

        $similar_text_b = (100.0 - $percent) * 1.0 * strlen($string_a.$string_b);

        return ($similar_text_a + $similar_text_b) / 200.0;

}

function clean_string($result_value){

        $result_value = preg_replace('~[\r\n]+~', '', $result_value);
        $result_value = preg_replace('~[\t]+~', '', $result_value);
```

```
        if(strlen($result_value) > 0){

                $result_value = str_replace(' ', ' ', $result_value);

                if(strlen($result_value) > 0){

                        $result_value = str_replace('nbsp;', ' ', $result_value);

                }

        }

        while(substr_count($result_value, '  ') > 0){

                $result_value = str_replace('  ', ' ', $result_value);

        }

        $result_value = str_replace('&gt;', '>', $result_value);
        $result_value = str_replace('&lt;', '<', $result_value);

        return $result_value;

}

function data_search($text,$start,$end){
```

```
$start_len = strlen($start);

$pos1 = strpos($text, $start) + $start_len;

$cut_page = substr($text,$pos1);

$pos2 = strpos($cut_page, $end);

$data = substr($cut_page,0,$pos2);

return $data;

}

function file_scrape($url){

        $html_body = file_get_contents($url);
        $dom = new simple_html_dom(null);
        $dom->load($html_body);
        $html = $dom;

        $result_table = array();



        if(true){
```

```
#                   $package_url_table = ;

                    foreach($html->find('body table.overviewSummary')[0]->find('a') as $url){

                            if(strpos($url, 'package-summary') !== false){

                                    $url_arr[] = array($url->href, $url->plaintext);

                            }

                    }
                    /*
                    $package_url_tbody = $package_url_table->find('tbody')[1];

                    echo $package_url_table;

                    foreach($package_url_tbody as $package_url_tr){

                            $package_url_td = $package_url_tr->find('td')[0];
                            $package_url = $package_url_td->find('a')[0]->href;

                            echo $package_url;

                    }
                    */

            }

            return $url_arr;
```

```
}



function scrape($url, $name, $fp, $test, $url_stub){

        $url = $url_stub.$url;

        $html_body = file_get_contents($url);
        $dom = new simple_html_dom(null);
        $dom->load($html_body);
        $html = $dom;

        $result_table = array();



        if(true){


                $test_limit = 1;
                $test_count = 0;

                foreach($html->find('body div.contentContainer ul.blockList li') as $li){

                        $this_table_title = "";
```

```
if(strlen($li->find('table caption span')[0]->plaintext) > 0){

        $this_table_title = str_replace(' Summary', '', $li->find('table caption span')[0]->plaintext);

}

if(strlen($this_table_title) > 0){

        $this_table_title = str_replace(' ', '', $this_table_title);

}



foreach($li->find('table tbody tr td.colFirst a') as $url_html){

        $class_url = $url_html->href;
        $class_name = $url_html->plaintext;
        $class_url = str_replace('../../', '', $url_html->href);
        $class_url = str_replace('../', '', $class_url);
        $class_url = str_replace('../', '', $class_url);
        $class_url = str_replace('../', '', $class_url);
        $class_url = str_replace('../', '', $class_url);
        $class_url = str_replace('../', '', $class_url);
        $class_url = str_replace('../', '', $class_url);
        $class_url = str_replace('../', '', $class_url);

        $java_substr = 'java/';
```

```
$javax_substr = 'javax/';

$org_substr = 'org/';


$java_pos = 0;

$javax_pos = 0;

$org_pos = 0;


if(strpos($class_url, $java_substr) !== false){


        $java_pos = strpos($class_url, $java_substr);


}


if(strpos($class_url, $javax_substr) !== false){


        $javax_pos = strpos($class_url, $javax_substr);


}


if(strpos($class_url, $org_substr) !== false){


        $org_pos = strpos($class_url, $org_substr);


}


$class_pos = 0;


$class_pos_arr = array();
```

```
$class_pos_arr[] = $java_pos;

$class_pos_arr[] = $javax_pos;

$class_pos_arr[] = $org_pos;


#var_dump($class_pos_arr);


#echo sizeof($class_pos_arr);


$class_pos_value_arr = array();


foreach($class_pos_arr as $this_class_pos){


        if($this_class_pos > 0){


                $class_pos_value_arr[] = $this_class_pos;


        }


}


$class_pos_min = 0;


if(sizeof($class_pos_value_arr) > 1){


        $class_pos_min = min($class_pos_value_arr);


        #var_dump(sizeof($class_pos_value_arr))."bajs";


}else{
```

```
                if(sizeof($class_pos_value_arr) > 0){

                        $class_pos_min = intval($class_pos_value_arr[0]);



                }
}

if($class_pos_min > 0){

                $class_pos = $class_pos_min;

}



$class_url = substr($class_url, $class_pos, strlen($class_url));

$class_url = $url_stub.$class_url;

$temp_arr = array($name, $url, $this_table_title, $class_name, $class_url);

#var_dump($temp_arr);

#var_dump($temp_arr);

$nested_class_arr = class_scrape($class_url, $class_name, $temp_arr, $fp, $test, $url_stub);
```

```
foreach($nested_class_arr as $nested_class){

$nested_class_url = $nested_class[0];
$nested_class_name = $nested_class[1];

$nested_temp_arr = array($name, $url, $this_table_title, $nested_class_name,
$nested_class_url);

$nested_class_arr_2 = class_scrape($nested_class_url, $nested_class_name,
$nested_temp_arr, $fp, $test, $url_stub);

if(sizeof($nested_class_arr_2) > 0){

#echo 'nested_nested';

foreach($nested_class_arr_2 as $nested_class_2){

$nested_class_url_2 = $nested_class_2[0];
$nested_class_name_2 = $nested_class_2[1];

$nested_temp_arr_2 = array($name, $url, $this_table_title,
$nested_class_name_2, $nested_class_url_2);

$nested_class_arr_3 = class_scrape($nested_class_url_2,
$nested_class_name_2, $nested_temp_arr_2, $fp, $test, $url_stub);

if(sizeof($nested_class_arr_3) > 0){

echo 'nested_nested_nested';

# there are 4 cases of this
```

```
                                                                    }

                                                                }

                                                            }

                                                        }

                                                    }

                                                }

                                            }

                                        }

                                    }

}

function class_scrape($url, $name, $temp_arr, $fp, $test, $url_stub){

        $class_section = "";

        $method_name = "";

        $method_full = "";
```

```
$method_para = "";


$package_name = $temp_arr[0];

$version = $url_stub;

$package_section = $temp_arr[2];

$class_name = $temp_arr[3];


$count_add = 0;


$nested_url_arr = array();


$result_table = array();

$data_table = array();

$class_table = array();


$print_table = array();


$inside_table = false;


if(strpos(strtolower($url), strtolower('applet/Applet.html')) !== false && $test || !$test){


        $html_body = file_get_contents($url);




        #$table_start_remove = '<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0"
SUMMARY="">'."\n".'<TR ALIGN="right" VALIGN="">'."\n".'<TD NOWRAP><FONT SIZE="-1">'."\n".'<CODE>';

        #$table_end_remove = '</TD>'."\n".'</TR>'."\n".'</TABLE>'."\n".'</CODE>';


        #$html_body = str_replace($table_start_remove,'',$html_body);
```

```
#$html_body = str_replace($table_end_remove,'</CODE>',$html_body);



$dom = new simple_html_dom(null);
$dom->load($html_body);
$html = $dom;

#echo $html;



if(true){



    $class_full = "";

    $class_full = $html->find('div.contentContainer div.description ul.blockList li.blockList pre')[0]->plaintext;



    foreach($html->find('div.contentContainer div.details ul.blockList li.blockList ul.blockList h3') as $section_detail){

        $class_section = str_replace(' Detail', '', $section_detail->plaintext);

        foreach($section_detail->parent()->find('ul.blockList li.blockList') as $method_detail){
```

```
#echo $method_detail;

#$method_dom = new simple_html_dom(null);
#$method_dom->load($method_detail);

#echo $method_dom;

$method_full = $method_detail->find('pre')[0]->plaintext;
$method_name = $method_detail->find('h4')[0]->plaintext;



$method_para = "";

if(strtolower($class_section) == 'field'){

        $package_name_url = str_replace('.', '/', $package_name);

        $class_name_url = $class_name;

        if(strpos($class_name, '.') !== false){

                $class_name_url = explode('.', $class_name)[0];

        }

        $url_stub_url = str_replace('C:/doc', 'C:/jdk-source-doc', $url_stub);

        $url_stub_url = str_replace('/doc/', '/', $url_stub_url);
```

```
$source_file_url = $url_stub_url.$package_name_url.'/'.$class_name_url.'.java';

$source_file = file($source_file_url);

$class_source_lines = array();
$method_source_lines = array();

$class_found = false;

foreach($source_file as $source_row){

        if(strpos($source_row, $class_name_url) !== false){

                $class_found = true;

        }

        if($class_found == true){

                #$class_source_lines[] = $source_row;

                if(strpos($source_row, ' '.$method_name) !== false){

                        $method_source_lines[] = $source_row;

                }else{
```

```
                                        #echo 'could not find method line';

                                        #var_dump($source_row);
                                        #var_dump($method_name);

                                    }

                                }

                            }

                            if(sizeof($method_source_lines) == 0){

                                #echo 'could not find method';

#var_dump($source_file_url.':'.$package_name.'@'.$class_name.'#'.$method_name);

                            }

                            $method_declaration_line = "";

                            if(sizeof($method_source_lines) == 1){

                                $method_declaration_line = $method_source_lines[0];

                            }else{

                                #echo "bajs";
```

```
$min_key = 9999;
$min_key_2 = 9999;

$method_equals_source_lines = array();

$key_arr = array();
$key_arr_2 = array();

foreach($method_source_lines as
$this_method_source_line){

        if(strpos($this_method_source_line, ";") !==
false){

            $key = text_similarity($method_name,
$this_method_source_line);

            if(!isset($key_arr[$key])){

                $key_arr[$key] =
$this_method_source_line;

                if($key < $min_key){

                    $min_key = $key;

                }

            }else{
```

```
                    #echo 'two declaration lines
have the same similarity';

                }

                if(sizeof($key_arr) > 0){

                    $method_declaration_line =
$key_arr[$min_key];

                    if(!isset($key_arr[$min_key])){

                        #var_dump($key_arr);

                    }

                }

                if(strpos($this_method_source_line,
"=") !== false){

                    $method_equals_source_lines[] = $this_method_source_line;

                }

                if(sizeof($method_equals_source_lines)
== 1){

                    $method_declaration_line =
$method_equals_source_lines[0];
```

```
                                                                    }else{

            foreach($method_equals_source_lines as $this_method_equals_line){

                                                                        $key_2 =
text_similarity($method_name, $this_method_equals_line);

            if(!isset($key_arr_2[$key_2])){

            $key_arr_2[$key_2] = $this_method_equals_line;

                                                                            if($key_2
< $min_key_2){

            $min_key_2 = $key_2;

                                                                            }

                                                                        }else{

                                                                            #echo
'two declaration lines have the same similarity at equals level';

                                                                        }

                                                                        if(sizeof($key_arr_2)
> 0){
```

```
$method_declaration_line = $key_arr_2[$min_key_2];

                                                                    }

                                                                }

                                                            }

                                                        }

                                                    }

                                                }

                                            foreach($method_source_lines as
$this_method_source_line_2){

                                                $method_full_last_call = str_replace(' ', '',
clean_string($method_full));

                                                $this_method_last_call = str_replace(' ', '',
clean_string($this_method_source_line_2));

                                                if(strpos($this_method_last_call,
$method_full_last_call) !== false){

                                                    $method_declaration_line =
$this_method_source_line_2;
```

```
                                        #echo "good";

                                }

                        }



}

#echo $method_declaration_line;

if(strpos($method_declaration_line, '=') !== false){

        $eq_pos = strpos($method_declaration_line, '=');

        $decl_eq = substr($method_declaration_line, $eq_pos,
strlen($method_declaration_line));

        $semi_pos = strpos($decl_eq, ';');

        $decl_semi = substr($decl_eq, 0, $semi_pos);

        $para = "";

        if(strpos($decl_semi, '= ') !== false){

                $para = str_replace('= ', '', $decl_semi);
```

```
                                        }else{

                                                $para = str_replace('=', '', $decl_semi);

                                        }

                                        #var_dump($para);


                                        $method_para = $para;

                                }

                        }


                                        $result_row = array($package_name, $package_section, $class_name, $class_full,
$class_section, $method_name, $method_full, $method_para, $version);

                                        $print_row = array();

                                        foreach($result_row as $result_value){

                                                #$result_value = preg_replace ('%[\x00-\x30\x127]%', '',
$result_value);
                                                $result_value = preg_replace('~[\r\n]+~', '', $result_value);
                                                $result_value = preg_replace('~[\t]+~', '', $result_value);
```

```
                    if(strlen($result_value) > 0){

                        $result_value = str_replace(' ', ' ', $result_value);

                        if(strlen($result_value) > 0){

                            $result_value = str_replace('nbsp;', ' ',
$result_value);

                        }

                    }

                    while(substr_count($result_value, '  ') > 0){

                        $result_value = str_replace('  ', ' ', $result_value);

                    }

                    $result_value = str_replace('&gt;', '>', $result_value);
                    $result_value = str_replace('&lt;', '<', $result_value);




                    $print_row[] = $result_value;

                }

                $print_table[] = $print_row;
```

```
                $count_add++;

        }

        foreach($section_detail->parent()->find('ul.blockListLast li.blockList') as $method_detail){

                #echo $method_detail;

                #$method_dom = new simple_html_dom(null);
                #$method_dom->load($method_detail);

                #echo $method_dom;

                $method_full = $method_detail->find('pre')[0]->plaintext;
                $method_name = $method_detail->find('h4')[0]->plaintext;



                $method_para = "";

                if(strtolower($class_section) == 'field'){

                        $package_name_url = str_replace('.', '/', $package_name);

                        $class_name_url = $class_name;

                        if(strpos($class_name, '.') !== false){
```

```
                $class_name_url = explode('.', $class_name)[0];

        }

        $url_stub_url = str_replace('C:/doc', 'C:/jdk-source-doc', $url_stub);

        $url_stub_url = str_replace('/doc/', '/', $url_stub_url);

        $source_file_url =
$url_stub_url.$package_name_url.'/'.$class_name_url.'.java';

        $source_file = file($source_file_url);

        $class_source_lines = array();
        $method_source_lines = array();

        $class_found = false;

        foreach($source_file as $source_row){

                if(strpos($source_row, $class_name_url) !== false){

                        $class_found = true;

                }

                if($class_found == true){

                        #$class_source_lines[] = $source_row;
```

```
                                        if(strpos($source_row, ' '.$method_name) !==
false){

                                                $method_source_lines[] = $source_row;

                                        }else{

                                                #echo 'could not find method line';

                                                #var_dump($source_row);

                                                #var_dump($method_name);

                                        }

                                }

                        }

                        if(sizeof($method_source_lines) == 0){

                                #echo 'could not find method';

        #var_dump($source_file_url.':'.$package_name.'@'.$class_name.'#'.$method_name);

                        }

                        $method_declaration_line = "";
```

```
if(sizeof($method_source_lines) == 1){

        $method_declaration_line = $method_source_lines[0];

}else{

        #echo "bajs";

        $min_key = 9999;
        $min_key_2 = 9999;

        $method_equals_source_lines = array();

        $key_arr = array();
        $key_arr_2 = array();

        foreach($method_source_lines as
$this_method_source_line){

                if(strpos($this_method_source_line, ";") !==
false){

                        $key = text_similarity($method_name,
$this_method_source_line);

                        if(!isset($key_arr[$key])){

                                $key_arr[$key] =
$this_method_source_line;

                                if($key < $min_key){
```

```
                                        $min_key = $key;

                                    }

                                }else{

                                    #echo 'two declaration lines
have the same similarity';

                                }

                                if(sizeof($key_arr) > 0){

                                    $method_declaration_line =
$key_arr[$min_key];

                                    if(!isset($key_arr[$min_key])){

                                        #var_dump($key_arr);

                                    }

                                }

                                if(strpos($this_method_source_line,
"=") !== false){

                                    $method_equals_source_lines[] = $this_method_source_line;
```

```
                                                                    }

                                                                    if(sizeof($method_equals_source_lines)
== 1){

                                                                        $method_declaration_line =
$method_equals_source_lines[0];

                                                                    }else{

            foreach($method_equals_source_lines as $this_method_equals_line){

                                                                        $key_2 =
text_similarity($method_name, $this_method_equals_line);

            if(!isset($key_arr_2[$key_2])){

            $key_arr_2[$key_2] = $this_method_equals_line;

                                                                        if($key_2
< $min_key_2){

            $min_key_2 = $key_2;

                                                                        }

                                                                    }else{
```

```
                                                                                #echo
'two declaration lines have the same similarity at equals level';

                                                                                }

                                                                                if(sizeof($key_arr_2)
> 0){

        $method_declaration_line = $key_arr_2[$min_key_2];

                                                                                }

                                                                        }

                                                                }

                                                        }

                                                }

                                                foreach($method_source_lines as
$this_method_source_line_2){
```

```
clean_string($method_full));

clean_string($this_method_source_line_2));


$method_full_last_call) !== false){

$this_method_source_line_2;


strlen($method_declaration_line));
```

```
                $method_full_last_call = str_replace(' ', '',

                $this_method_last_call = str_replace(' ', '',


                if(strpos($this_method_last_call,

                        $method_declaration_line =

                        #echo "good";

                }

        }


}

#echo $method_declaration_line;

if(strpos($method_declaration_line, '=') !== false){

        $eq_pos = strpos($method_declaration_line, '=');

        $decl_eq = substr($method_declaration_line, $eq_pos,

        $semi_pos = strpos($decl_eq, ';');
```

```
                $decl_semi = substr($decl_eq, 0, $semi_pos);

                $para = "";

                if(strpos($decl_semi, '= ') !== false){

                        $para = str_replace('= ', '', $decl_semi);

                }else{

                        $para = str_replace('=', '', $decl_semi);

                }

                #var_dump($para);

                $method_para = $para;

            }

        }

                $result_row = array($package_name, $package_section, $class_name, $class_full,
$class_section, $method_name, $method_full, $method_para, $version);

                $print_row = array();
```

```
foreach($result_row as $result_value){

#$result_value = preg_replace ('%[\x00-\x30\x127]%', '',
$result_value);

$result_value = preg_replace('~[\r\n]+~', '', $result_value);

$result_value = preg_replace('~[\t]+~', '', $result_value);

if(strlen($result_value) > 0){

$result_value = str_replace(' ', ' ', $result_value);

if(strlen($result_value) > 0){

$result_value = str_replace('nbsp;', ' ',
$result_value);

}

}

while(substr_count($result_value, '  ') > 0){

$result_value = str_replace('  ', ' ', $result_value);

}

$result_value = str_replace('&gt;', '>', $result_value);

$result_value = str_replace('&lt;', '<', $result_value);
```

```
                                        $print_row[] = $result_value;

                                    }

                                    $print_table[] = $print_row;



                                    $count_add++;

                                }

                            }

                            #echo $html;

                                foreach($html->find('div.contentContainer div.summary ul.blockList li.blockList ul.blockList
li.blockList h3') as $summary_section){

                                    #echo $summary_section;

                                    if(strpos(strtolower($summary_section->plaintext), 'summary') !== false){
```

```
                                    if(strpos(strtolower($summary_section->plaintext), 'nested') !== false){

                                        #foreach($summary_section->parent()->find('table tbody tr td.colLast
code span a') as $nested_link){

                                        foreach($summary_section->parent()->find('table tbody tr td.colLast
code') as $nested_link){

                                            $dom_nested = new simple_html_dom(null);
                                            $dom_nested->load($nested_link->innertext);

                                            foreach($dom_nested->find('a') as $nested_href){

                                                $nested_name = $nested_href->plaintext;

                                                $nested_url = str_replace('../../', '', $nested_href-
>href);
                                                $nested_url = str_replace('../', '', $nested_url);
                                                $nested_url = str_replace('../', '', $nested_url);
                                                $nested_url = str_replace('../', '', $nested_url);
                                                $nested_url = str_replace('../', '', $nested_url);
                                                $nested_url = str_replace('../', '', $nested_url);
                                                $nested_url = str_replace('../', '', $nested_url);
                                                $nested_url = str_replace('../', '', $nested_url);

                                                $java_substr = 'java/';
                                                $javax_substr = 'javax/';
                                                $org_substr = 'org/';

                                                $java_pos = 0;
```

```
$javax_pos = 0;
$org_pos = 0;

if(strpos($nested_url, $java_substr) !== false){
        $java_pos = strpos($nested_url,
$java_substr);
}

if(strpos($nested_url, $javax_substr) !== false){
        $javax_pos = strpos($nested_url,
$javax_substr);
}

if(strpos($nested_url, $org_substr) !== false){
        $org_pos = strpos($nested_url,
$org_substr);
}

$nested_pos = 0;

$nested_pos_arr = array();

$nested_pos_arr[] = $java_pos;
$nested_pos_arr[] = $javax_pos;
$nested_pos_arr[] = $org_pos;
```

```
#var_dump($class_pos_arr);

#echo sizeof($class_pos_arr);

$nested_pos_value_arr = array();

foreach($nested_pos_arr as $this_nested_pos){

        if($this_nested_pos > 0){

                $nested_pos_value_arr[] =
$this_nested_pos;

        }

}

$nested_pos_min = 0;

if(sizeof($nested_pos_value_arr) > 1){

        $nested_pos_min =
min($nested_pos_value_arr);

        #var_dump(sizeof($class_pos_value_arr))."bajs";

}else{

        if(sizeof($nested_pos_value_arr) > 0){
```

```
                                                    $nested_pos_min =
intval($nested_pos_value_arr[0]);

                                                }
                                            }

                                            if($nested_pos_min > 0){

                                                $nested_pos = $nested_pos_min;

                                            }

                                                    $nested_url = substr($nested_url, $nested_pos,
strlen($nested_url));

                                                    $nested_url = $url_stub.$nested_url;

                                                    $nested_url_arr[] = array($nested_url,
$nested_name);

                                        }
```

```
                                                    }

                                                }

                                            }

                                        }

                                }

                        }

                }
```

```
        if($count_add == 0){

                $result_row = array($package_name, $package_section, $class_name, $class_full, $class_section, $method_name, $method_full, $method_para, $version);

                $print_row = array();

                foreach($result_row as $result_value){
```

```
#$result_value = preg_replace ('%[\x00-\x30\x127]%', '', $result_value);
$result_value = preg_replace('~[\r\n]+~', '', $result_value);
$result_value = preg_replace('~[\t]+~', '', $result_value);

if(strlen($result_value) > 0){

        $result_value = str_replace(' ', ' ', $result_value);

        if(strlen($result_value) > 0){

                $result_value = str_replace('nbsp;', ' ', $result_value);

        }

}

while(substr_count($result_value, '  ') > 0){

        $result_value = str_replace('  ', ' ', $result_value);

}

$result_value = str_replace('&gt;', '>', $result_value);
$result_value = str_replace('&lt;', '<', $result_value);



$print_row[] = $result_value;
```

```
                }

                $print_table[] = $print_row;

        }


        foreach($print_table as $this_row){

                fputcsv($fp, $this_row);

        }


        return $nested_url_arr;

}


#error_reporting(0);


$header_array = array("package_name", "package_url", "package_section", "class_name", "class_url", "class_section", "jdk_version",
"method_full", "method_type", "method_name", "method", "parameters", "method_version", "method_url");
```

```
$header_array = array("package_name", "package_section", "class_name", "class_full", "class_section", "method_name",
"method_full", "method_para", "version");


$scrape_file = "final7.csv";


if(!file_exists($scrape_file)){


        $fp = fopen($scrape_file, 'a');


   fputcsv($fp, $header_array);


}else{


        $fp = fopen($scrape_file, 'a');


}


$test = false;


#check links for 'package-summary'


#$file_arr[] = 'C:/doc/1.0/doc/';
```

```
$file_arr[] = 'C:/doc/1.0/doc/';

$file_arr[] = 'C:/doc/1.1/doc/';

$file_arr[] = 'C:/doc/1.2/doc/';

$file_arr[] = 'C:/doc/1.3/doc/';

$file_arr[] = 'C:/doc/1.4/doc/';

$file_arr[] = 'C:/doc/1.5/doc/';

$file_arr[] = 'C:/doc/1.6/doc/';

$file_arr[] = 'C:/doc/1.7/doc/';

$file_arr[] = 'C:/doc/1.8/doc/';









for($j = 0; $j < sizeof($file_arr); $j++){

        $url_arr = array();

        $url_arr = file_scrape($file_arr[$j].'overview-summary.html');


        foreach($url_arr as $url){


                $package_url = $url[0];

                $package_name = $url[1];


                #var_dump(array($package_url, $package_name));


                scrape($package_url, $package_name, $fp, $test, $file_arr[$j]);


        }
```

```
}

fclose($fp);

?>
</pre>
```

APPENDIX G – Evolution of Java API Packages

Java API Packages JDK Versions 1-8

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
sun.tools.debug	X								
java.awt.peer	X								
java.applet	X	X	X	X	X	X	X	X	X
java.awt.image	X	X	X	X	X	X	X	X	X
java.awt	X	X	X	X	X	X	X	X	X
java.net	X	X	X	X	X	X	X	X	X
java.util	X	X	X	X	X	X	X	X	X
java.io	X	X	X	X	X	X	X	X	X
java.lang	X	X	X	X	X	X	X	X	X
java.awt.datatransfer		X	X	X	X	X	X	X	X
java.awt.event		X	X	X	X	X	X	X	X
java.beans		X	X	X	X	X	X	X	X
java.lang.reflect		X	X	X	X	X	X	X	X
java.math		X	X	X	X	X	X	X	X
java.rmi		X	X	X	X	X	X	X	X
java.rmi.dgc		X	X	X	X	X	X	X	X
java.rmi.registry		X	X	X	X	X	X	X	X
java.rmi.server		X	X	X	X	X	X	X	X
java.security		X	X	X	X	X	X	X	X
java.security.acl		X	X	X	X	X	X	X	X
java.security.interfaces		X	X	X	X	X	X	X	X
java.sql		X	X	X	X	X	X	X	X
java.text		X	X	X	X	X	X	X	X
java.util.zip		X	X	X	X	X	X	X	X
java.awt.color			X	X	X	X	X	X	X
java.awt.dnd			X	X	X	X	X	X	X
java.awt.font			X	X	X	X	X	X	X
java.awt.geom			X	X	X	X	X	X	X
java.awt.im			X	X	X	X	X	X	X
java.awt.image.renderable			X	X	X	X	X	X	X
java.awt.print			X	X	X	X	X	X	X
java.beans.beancontext			X	X	X	X	X	X	X
java.lang.ref			X	X	X	X	X	X	X
java.rmi.activation			X	X	X	X	X	X	X
java.security.cert			X	X	X	X	X	X	X
java.security.spec			X	X	X	X	X	X	X
java.util.jar			X	X	X	X	X	X	X
javax.accessibility			X	X	X	X	X	X	X
javax.swing			X	X	X	X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
javax.swing.border			X	X	X	X	X	X	X
javax.swing.colorchooser			X	X	X	X	X	X	X
javax.swing.event			X	X	X	X	X	X	X
javax.swing.filechooser			X	X	X	X	X	X	X
javax.swing.plaf			X	X	X	X	X	X	X
javax.swing.plaf.basic			X	X	X	X	X	X	X
javax.swing.plaf.metal			X	X	X	X	X	X	X
javax.swing.plaf.multi			X	X	X	X	X	X	X
javax.swing.table			X	X	X	X	X	X	X
javax.swing.text			X	X	X	X	X	X	X
javax.swing.text.html			X	X	X	X	X	X	X
javax.swing.text.html.parser			X	X	X	X	X	X	X
javax.swing.text.rtf			X	X	X	X	X	X	X
javax.swing.tree			X	X	X	X	X	X	X
javax.swing.undo			X	X	X	X	X	X	X
org.omg.CORBA			X	X	X	X	X	X	X
org.omg.CORBA.DynAnyPackage			X	X	X	X	X	X	X
org.omg.CORBA.ORBPackage			X	X	X	X	X	X	X
org.omg.CORBA.portable			X	X	X	X	X	X	X
org.omg.CORBA.TypeCodePackage			X	X	X	X	X	X	X
org.omg.CosNaming			X	X	X	X	X	X	X
org.omg.CosNaming.NamingContextPackage			X	X	X	X	X	X	X
java.awt.im.spi				X	X	X	X	X	X
javax.naming				X	X	X	X	X	X
javax.naming.directory				X	X	X	X	X	X
javax.naming.event				X	X	X	X	X	X
javax.naming.ldap				X	X	X	X	X	X
javax.naming.spi				X	X	X	X	X	X
javax.rmi				X	X	X	X	X	X
javax.rmi.CORBA				X	X	X	X	X	X
javax.sound.midi				X	X	X	X	X	X
javax.sound.midi.spi				X	X	X	X	X	X
javax.sound.sampled				X	X	X	X	X	X
javax.sound.sampled.spi				X	X	X	X	X	X
javax.transaction				X	X	X	X	X	X
org.omg.CORBA_2_3				X	X	X	X	X	X
org.omg.CORBA_2_3.portable				X	X	X	X	X	X
org.omg.SendingContext				X	X	X	X	X	X
org.omg.stub.java.rmi				X	X	X	X	X	X
java.nio					X	X	X	X	X
java.nio.channels					X	X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
java.nio.channels.spi					X	X	X	X	X
java.nio.charset					X	X	X	X	X
java.nio.charset.spi					X	X	X	X	X
java.util.logging					X	X	X	X	X
java.util.prefs					X	X	X	X	X
java.util.regex					X	X	X	X	X
javax.crypto					X	X	X	X	X
javax.crypto.interfaces					X	X	X	X	X
javax.crypto.spec					X	X	X	X	X
javax.imageio					X	X	X	X	X
javax.imageio.event					X	X	X	X	X
javax.imageio.metadata					X	X	X	X	X
javax.imageio.plugins.jpeg					X	X	X	X	X
javax.imageio.spi					X	X	X	X	X
javax.imageio.stream					X	X	X	X	X
javax.net					X	X	X	X	X
javax.net.ssl					X	X	X	X	X
javax.print					X	X	X	X	X
javax.print.attribute					X	X	X	X	X
javax.print.attribute.standard					X	X	X	X	X
javax.print.event					X	X	X	X	X
javax.security.auth					X	X	X	X	X
javax.security.auth.callback					X	X	X	X	X
javax.security.auth.kerberos					X	X	X	X	X
javax.security.auth.login					X	X	X	X	X
javax.security.auth.spi					X	X	X	X	X
javax.security.auth.x500					X	X	X	X	X
javax.security.cert					X	X	X	X	X
javax.sql					X	X	X	X	X
javax.transaction.xa					X	X	X	X	X
javax.xml.parsers					X	X	X	X	X
javax.xml.transform					X	X	X	X	X
javax.xml.transform.dom					X	X	X	X	X
javax.xml.transform.sax					X	X	X	X	X
javax.xml.transform.stream					X	X	X	X	X
org.ietf.jgss					X	X	X	X	X
org.omg.CosNaming.NamingContextExtPackage					X	X	X	X	X
org.omg.Dynamic					X	X	X	X	X
org.omg.DynamicAny					X	X	X	X	X
org.omg.DynamicAny.DynAnyFactoryPackage					X	X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
org.omg.DynamicAny.DynAnyPackage					X	X	X	X	X
org.omg.IOP					X	X	X	X	X
org.omg.IOP.CodecFactoryPackage					X	X	X	X	X
org.omg.IOP.CodecPackage					X	X	X	X	X
org.omg.Messaging					X	X	X	X	X
org.omg.PortableInterceptor					X	X	X	X	X
org.omg.PortableInterceptor.ORBInitInfo Package					X	X	X	X	X
org.omg.PortableServer					X	X	X	X	X
org.omg.PortableServer.CurrentPackage					X	X	X	X	X
org.omg.PortableServer.POAManagerPac kage					X	X	X	X	X
org.omg.PortableServer.POAPackage					X	X	X	X	X
org.omg.PortableServer.portable					X	X	X	X	X
org.omg.PortableServer.ServantLocatorP ackage					X	X	X	X	X
org.w3c.dom					X	X	X	X	X
org.xml.sax					X	X	X	X	X
org.xml.sax.ext					X	X	X	X	X
org.xml.sax.helpers					X	X	X	X	X
java.lang.annotation						X	X	X	X
java.lang.instrument						X	X	X	X
java.lang.management						X	X	X	X
java.util.concurrent						X	X	X	X
java.util.concurrent.atomic						X	X	X	X
java.util.concurrent.locks						X	X	X	X
javax.imageio.plugins.bmp						X	X	X	X
javax.management						X	X	X	X
javax.management.loading						X	X	X	X
javax.management.modelmbean						X	X	X	X
javax.management.monitor						X	X	X	X
javax.management.openmbean						X	X	X	X
javax.management.relation						X	X	X	X
javax.management.remote						X	X	X	X
javax.management.remote.rmi						X	X	X	X
javax.management.timer						X	X	X	X
javax.rmi.ssl						X	X	X	X
javax.security.sasl						X	X	X	X
javax.sql.rowset						X	X	X	X
javax.sql.rowset.serial						X	X	X	X
javax.sql.rowset.spi						X	X	X	X
javax.swing.plaf.synth						X	X	X	X
javax.xml						X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
javax.xml.datatype						X	X	X	X
javax.xml.namespace						X	X	X	X
javax.xml.validation						X	X	X	X
javax.xml.xpath						X	X	X	X
org.w3c.dom.bootstrap						X	X	X	X
org.w3c.dom.events						X	X	X	X
org.w3c.dom.ls						X	X	X	X
javax.activity						X	X	X	X
java.text.spi							X	X	X
java.util.spi							X	X	X
javax.activation							X	X	X
javax.annotation							X	X	X
javax.annotation.processing							X	X	X
javax.jws							X	X	X
javax.jws.soap							X	X	X
javax.lang.model							X	X	X
javax.lang.model.element							X	X	X
javax.lang.model.type							X	X	X
javax.lang.model.util							X	X	X
javax.script							X	X	X
javax.tools							X	X	X
javax.xml.bind							X	X	X
javax.xml.bind.annotation							X	X	X
javax.xml.bind.annotation.adapters							X	X	X
javax.xml.bind.attachment							X	X	X
javax.xml.bind.helpers							X	X	X
javax.xml.bind.util							X	X	X
javax.xml.crypto							X	X	X
javax.xml.crypto.dom							X	X	X
javax.xml.crypto.dsig							X	X	X
javax.xml.crypto.dsig.dom							X	X	X
javax.xml.crypto.dsig.keyinfo							X	X	X
javax.xml.crypto.dsig.spec							X	X	X
javax.xml.soap							X	X	X
javax.xml.stream							X	X	X
javax.xml.stream.events							X	X	X
javax.xml.stream.util							X	X	X
javax.xml.transform.stax							X	X	X
javax.xml.ws							X	X	X
javax.xml.ws.handler							X	X	X
javax.xml.ws.handler.soap							X	X	X
javax.xml.ws.http							X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
javax.xml.ws.soap							X	X	X
javax.xml.ws.spi							X	X	X
javax.xml.ws.wsaddressing							X	X	X
java.lang.invoke								X	X
java.nio.file								X	X
java.nio.file.attribute								X	X
java.nio.file.spi								X	X
javax.swing.plaf.nimbus								X	X
javax.xml.ws.spi.http								X	X
java.time									X
java.time.chrono									X
java.time.format									X
java.time.temporal									X
java.time.zone									X
java.util.function									X
java.util.stream									X
org.w3c.dom.views									X
Total	9	22	59	76	135	166	203	209	217

APPENDIX H – 37 Packages compared to Google lists of desired packages

37 Java Packages Ultimately Copied	3/28/2007 Noser Statement of Work	1/4/2007 Email from Dan Bornstein
java.awt.font		
java.beans		java.beans
java.io	java.io	java.io
java.lang	java.lang	java.lang
java.lang.annotation		
java.lang.ref	java.lang.ref	
java.lang.reflect	java.lang.reflect	
java.net	java.net	java.net
java.nio	java.nio	java.nio
java.nio.channels		
java.nio.channels.spi		
java.nio.charset		
java.nio.charset.spi		
java.security	java.security	java.security
java.security.acl	java.security.acl	
java.security.cert	java.security.cert	
java.security.interfaces	java.security.interfaces	
java.security.spec	java.security.spec	
java.sql	java.sql	java.sql
java.text	java.text	java.text
java.util	java.util	java.util
java.util.jar	java.util.jar	java.util.jar
java.util.logging	java.util.logging	java.util.logging
java.util.prefs	java.util.prefs	java.util.prefs
java.util.regex	java.util.regex	java.util.regex
java.util.zip	java.util.zip	java.util.zip
javax.crypto	javax.crypto	javax.crypto
javax.crypto.interfaces	javax.crypto.interfaces	
javax.crypto.spec	javax.crypto.spec	
javax.net	javax.net	javax.net
javax.net.ssl	javax.net.ssl	
javax.security.auth		
javax.security.auth.callback		
javax.security.auth.login		
javax.security.auth.x500		
javax.security.cert	javax.security.cert	
javax.sql	javax.sql	javax.sql
	java.math java.awt.image.renderabl java.awt java.awt.image javax.imageio javax.sound javax.xml javax.transaction	java.math java.awt java.util.concurrent javax.imageio javax.security javax.sound javax.xml java.rmi java.applet javax.accessibility javax.transaction javax.management javax.naming javax.print

APPENDIX I – Parser Scripts for Android Documentation

PHP Script to Parse Text Files from API Levels 14-23:

```
<pre>

<?php

function parse($url){

        $print_table = array();

        $file_array = array();

        $handle = fopen($url, "r");
        if($handle){
                while(($buffer = fgets($handle, 4096)) !== false){
                        array_push($file_array, $buffer);
                }
                if(!feof($handle)){
                        echo "Error: unexpected fgets() fail\n";
                }
                fclose($handle);
        }


        $location = 0;

        $this_arr = array();

        $package_locations = array();
```

```
$package_names = array();

foreach($file_array as $file_row){

        array_push($this_arr, substr($file_row, 0, 8));



        $package_len = strlen("package");

        $package_part = substr($file_row, 0, $package_len);

        if(strtolower($package_part) == 'package'){

                $package = str_replace("package ", "", $file_row);
                $this_package_name = str_replace(" {", "", $package);
                array_push($package_names, $this_package_name);

                array_push($print_table, array($this_package_name, $url));

                array_push($package_locations, $location);

        }

        $location++;
        #echo $first_part;

}
```

```
#print_r(array_unique($this_arr));

array_push($package_locations, $location);

#print_r($package_locations);

$this_package = array();

$final_package_table = array();
$final_class_table = array();
$final_method_table = array();



### package loop

for($i = 0; $i < sizeof($package_locations)-1; $i++){

        $package_name = $package_names[$i];

        array_push($final_package_table, array($package_name, $url));

        $start = $package_locations[$i]+1;
        $end = $package_locations[$i+1];

        for($j = $start; $j < $end; $j++){

                array_push($this_package, $file_array[$j]);
```

```
}

$this_package_count = 0;

$class_locations_start = array();
$class_locations_end = array();

foreach($this_package as $package_row){

        if(strpos($package_row, ' {')  !== false){

                array_push($class_locations_start, $this_package_count);

        }

        if(strpos($package_row, ' }')  !== false){

                array_push($class_locations_end, $this_package_count);

        }

        $this_package_count++;

}

#print_r($class_locations_end);


### class loop
```

```
for($k = 0; $k < sizeof($class_locations_start); $k++){

        $start_cl = $class_locations_start[$k];

        $end_cl = $class_locations_end[$k];




        $this_class = array();

        for($l = $start_cl; $l < $end_cl; $l++){

                array_push($this_class, $this_package[$l]);

        }

        if($k == 0){

        #print_r($this_class);

        }

        #

        $class_header_row = $this_class[0];

        $class = str_replace(" {", "", $class_header_row);

        $search_pos = -1;

        $search_pos_count = 0;
```

```
if(strpos($class, " implements") !== false && strpos($class, " extends") !== false){

        $search_pos = min(strpos($class, " implements"), strpos($class, " extends"));
        $search_pos_count = 1;

}else{

        if(strpos($class, " implements") !== false){


                ### need to create file


                $search_pos = strpos($class, " implements");
                $search_pos_count = 2;

        }else{

                if(strpos($class, " extends") !== false){


                        ## need to add to var


                        $search_pos = strpos($class, " extends");
                        $search_pos_count = 3;

                }

        }
```

```
}

$class_extends = "";

$class_extends_temp = "";

$class_implements = "";

$class_implements_temp = "";

# public class SharedPreferencesBackupHelper extends android.app.backup.FileBackupHelperBase implements android.app.backup.BackupHelper {

if($search_pos > 0 && $search_pos_count == 1){

    ## extends first
    if(strpos($class, " implements") > strpos($class, " extends")){

        $class_extends_temp = substr($class, strpos($class, " extends"), strpos($class, " implements") - strpos($class, " extends"));

        $class_extends_temp = explode(' ', $class_extends_temp);

        $class_extends_piece_count = 0;

        foreach($class_extends_temp as $class_extends_piece){

            if($class_extends_piece_count > 1){

                if(strlen($class_extends_piece) > 0){
```

```
                                        if(strlen($class_extends) < 1){

                                                $class_extends = $class_extends_piece;

                                        }else{

                                                $class_extends = $class_extends.",
".$class_extends_piece;

                                        }

                                }

                        }

                        $class_extends_piece_count++;

                }

                $class_implements_temp = substr($class, strpos($class, " implements"),
strlen($class) - strpos($class, " implements"));

                $class_implements_temp = explode(' ', $class_implements_temp);

                $class_implements_piece_count = 0;

                foreach($class_implements_temp as $class_implements_piece){

                        if($class_implements_piece_count > 1){
```

```
                                        if(strlen($class_implements_piece) > 0 &&
strpos($class_implements_piece, '{') == false){

                                            if(strlen($class_implements) < 1){

                                                $class_implements =
$class_implements_piece;

                                            }else{

                                                $class_implements =
$class_implements.", ".$class_implements_piece;

                                            }

                                        }

                                    }

                                    $class_implements_piece_count++;

                                }



                            }

                            ## implements first

                            if(strpos($class, " implements") < strpos($class, " extends")){
```

```
                                        $class_implements_temp = substr($class, strpos($class, " implements"),
strpos($class, " extends") - strpos($class, " implements"));

                                        $class_implements_temp = explode(' ', $class_implements_temp);

                                        $class_implements_piece_count = 0;

                                        foreach($class_implements_temp as $class_implements_piece){

                                                if($class_implements_piece_count > 1){

                                                        if(strlen($class_implements_piece) > 0){

                                                                if(strlen($class_implements) < 1){

                                                                        $class_implements =
$class_implements_piece;

                                                                }else{

                                                                        $class_implements =
$class_implements.", ".$class_implements_piece;

                                                                }

                                                        }

                                                }

                                                $class_implements_piece_count++;
```

```
}

$class_extends_temp = substr($class, strpos($class, " extends"), strlen($class) -
strpos($class, " extends"));

$class_extends_temp = explode(' ', $class_extends_temp);

$class_extends_piece_count = 0;

foreach($class_extends_temp as $class_extends_piece){

if($class_extends_piece_count > 1){

if(strlen($class_extends_piece) > 0 &&
strpos($class_extends_piece, '{') == false){

if(strlen($class_extends) < 1){

$class_extends = $class_extends_piece;

}else{

$class_extends = $class_extends.",
".$class_extends_piece;

}

}

}
```

```
                                        $class_extends_piece_count++;

                                    }

                                }

                            }

                            if($search_pos > 0 && $search_pos_count == 2){

                                $class_implements_temp = substr($class, $search_pos, strlen($class) - $search_pos);

                                $class_implements_temp = explode(' ', $class_implements_temp);

                                $class_implements_piece_count = 0;

                                foreach($class_implements_temp as $class_implements_piece){

                                    if($class_implements_piece_count > 1){

                                        if(strlen($class_implements_piece) > 0 &&
strpos($class_implements_piece, '{') == false){

                                            if(strlen($class_implements) < 1){

                                                $class_implements = $class_implements_piece;
```

```
                                                    }else{

                                                                $class_implements = $class_implements.",
".$class_implements_piece;

                                                    }

                                                }

                                            }

                                            $class_implements_piece_count++;

                                        }

                                    }

                                    if($search_pos > 0 && $search_pos_count == 3){

                                        $class_extends_temp = substr($class, $search_pos, strlen($class) - $search_pos);

                                        $class_extends_temp = explode(' ', $class_extends_temp);

                                        $class_extends_piece_count = 0;

                                        foreach($class_extends_temp as $class_extends_piece){

                                            if($class_extends_piece_count > 1){
```

```
                                        if(strlen($class_extends_piece) > 0 && strpos($class_extends_piece, '{')
== false){

                                            if(strlen($class_extends) < 1){

                                                $class_extends = $class_extends_piece;

                                            }else{

                                                $class_extends = $class_extends.",
".$class_extends_piece;

                                            }

                                        }

                                    }

                                    $class_extends_piece_count++;

                                }

                            }

                            #echo "extends ". $class_extends . " implements " . $class_implements . "\n\r";



                            if($search_pos > 0){
```

```
            $class = substr($class, 0, $search_pos);

}

#interface, class
$class_type = "";
$class_name = '';
$class_abstract = 0;
$class_static = 0;
$class_final = 0;
$class_deprecated = 0;
$class_visibility = "";

if(strpos($class, "class ") !== false){

            $class_type = "class";

}else{

            if(strpos($class, "interface ") !== false){

                        $class_type = "interface";

            }

}

if(strpos($class, "abstract ") !== false){
```

```
                $class_abstract = 1;

}

if(strpos($class, "static ") !== false){

                $class_static = 1;

}

if(strpos($class, "final ") !== false){

                $class_final = 1;

}

if(strpos($class, "deprecated ") !== false){

                $class_deprecated = 1;

}


if(strpos($class, "public ") !== false){

                $class_visibility = "public";

}else{
```

```
                if(strpos($class, "protected ") !== false){

                $class_visibility = "protected";

                }else{

                        if(strpos($class, "private ") !== false){

                        $class_visibility = "private";

                        }else{

                                $class_visibility = "no modifier";

                        }

                }

}









#var_dump($class);

$class = explode(' ', $class);
```

```
#var_dump($class);

$class_name = $class[sizeof($class)-1];

$dump = $package_name."@".$class_name."-".$url;

$dump = trim(preg_replace('/\s\s+/', ' ', $dump));

$dump = str_replace("n\\", '', $dump);
$dump = str_replace("r\\", '', $dump);
$dump = str_replace("t\\", '', $dump);
$dump = str_replace("n\r\\", '', $dump);
$dump = str_replace("r\n\\", '', $dump);
$dump = preg_replace('~[\r\n]+~', '', $dump);

#echo $dump."\n\r";

$method_text_array = array();

$header_count = 0;

foreach($this_class as $this_class_row){

        if($header_count > 0){

                if(strlen($this_class_row) > 2){
```

```
                                        array_push($method_text_array, $this_class_row);

                                }

                        }

                        $header_count++;

                }

                        $final_class_array = array($package_name, $class_type, $class_name, $class_abstract, $class_static,
$class_final, $class_deprecated, $class_visibility, $class_extends, $class_implements, $url);

                        array_push($final_class_table, $final_class_array);



                        ### method loop

                        foreach($method_text_array as $method_text){

                                $method_type = "";

                                #ctor, field, method, $enum_constant
                                $method_explode = explode(' ', $method_text);

                                $method_turned = 0;

                                $method_first_piece = "";

                                foreach($method_explode as $method_piece){
```

```
                if(strlen($method_piece) > 0 && $method_turned == 0){

                        $method_first_piece = $method_piece;

                        $method_turned = 1;

                }

        }

        if(strpos($method_first_piece, "enum_constant") !== false){

                $method_type = "enumerator value";

        }else{

                if(strpos($method_first_piece, "method") !== false){

                $method_type = "method";

                }else{

                        if(strpos($method_first_piece, "field") !== false){

                        $method_type = "field";

                        }else{

                                if(strpos($method_first_piece, "ctor") !== false){
```

```
                    $method_type = "constructor";

                    }else{

                        $method_type = "could not locate method type";

                    }

                }

            }

        }


#echo $method_type."\n\r";


$method_static = 0;
$method_final = 0;
$method_deprecated = 0;
$method_visibility = "";

if(strpos($method_text, "static ") !== false){

        $method_static = 1;

}
```

```
if(strpos($method_text, "final ") !== false){

        $method_final = 1;

}

if(strpos($method_text, "deprecated ") !== false){

        $method_deprecated = 1;

}

if(strpos($method_text, "public ") !== false){

$method_visibility = "public";

}else{

        if(strpos($method_text, "protected ") !== false){

        $method_visibility = "protected";

        }else{

                if(strpos($method_text, "private ") !== false){

                $method_visibility = "private";
```

```
                    }else{

                        $method_visibility = "no modifier";

                    }

                }

            }

            $method_return_type = '';
            $method_name = '';
            $method_transient = '';
            $method_volatile = '';
            $method_value = '';
            $method_parameter = '';

            if($method_type == 'field'){

                $method_text_temp = $method_text;

                $method_text_temp = explode(' ', $method_text_temp);

                $method_text_count = 0;
                $method_count_match = 0;

                foreach($method_text_temp as $method_temp_piece){

                    if(strpos(strtolower($method_temp_piece), "=") !== false &&
$method_count_match == 0){
```

```
                    $method_count_match = $method_text_count;

                    #echo $method_temp_piece;

                }

                $method_text_count++;

            }

            if($method_count_match >= 2){

                $method_return_type = $method_text_temp[$method_count_match-
2];
                $method_name = $method_text_temp[$method_count_match-1];
                $method_value = str_replace(';', '',
$method_text_temp[$method_count_match+1]);

            }else{

                $method_name = str_replace(';', '',
$method_text_temp[sizeof($method_text_temp)-1]);
                $method_return_type = str_replace(';', '',
$method_text_temp[sizeof($method_text_temp)-2]);

            }

            if(strpos($method_text, "transient ") !== false){

                $method_transient = 1;
```

```
                }else{

                        $method_transient = 0;

                }

                if(strpos($method_text, "volatile ") !== false){

                        $method_volatile = 1;

                }else{

                        $method_volatile = 0;

                }

                #echo $method_name;

        }

        $method_abstract = "";
        $method_synchronized = '';
        $method_exception = '';

        if($method_type == 'method'){

                $method_text_temp = $method_text;

                $method_text_temp = explode(' ', $method_text_temp);
```

```
$method_text_count = 0;
$method_count_match = 0;

foreach($method_text_temp as $method_temp_piece){

        if(strpos(strtolower($method_temp_piece), "(") !== false &&
$method_count_match == 0){

                $method_count_match = $method_text_count;

                #echo $method_temp_piece;

        }

        $method_text_count++;

}

$method_return_type = $method_text_temp[$method_count_match-1];
$method_name_temp = $method_text_temp[$method_count_match];

$method_name_temp = explode('(', $method_name_temp);
$method_name = $method_name_temp[0];

$method_parameter_temp = explode(')', $method_name_temp[1]);
$method_parameter = $method_parameter_temp[0];
#echo $method_parameter;
#echo $method_return_type;
```

```
if(strpos($method_text, "abstract ") !== false){

        $method_abstract = 1;

}else{

        $method_abstract = 0;

}

if(strpos($method_text, "synchronized ") !== false){

        $method_synchronized = 1;

}else{

        $method_synchronized = 0;

}


if(strpos($method_text, " throws ") !== false){

        $throws_pos = strpos($method_text, " throws ");

        if($throws_pos > 0){

                $method_throws_temp = substr($method_text,
$throws_pos, strlen($method_text) - $throws_pos);
```

```
                                                    #$method_throws_temp = explode(' ',
$method_throws_temp);

                                                    #$method_exception = str_replace(';', '',
$method_throws_temp[sizeof($method_throws_temp)-1]);

                                                    #echo $method_exception;

                                                    $method_throws_temp = str_replace(' throws ', '',
$method_throws_temp);

                                                    $method_exception = str_replace(';', '',
$method_throws_temp);

                                            }

                                    }



                            }

                            if($method_type == 'constructor'){

                                    $c_method_text_temp = $method_text;

                                    $c_method_text_temp = explode(' ', $c_method_text_temp);

                                    $c_method_text_count = 0;
                                    $c_method_count_match = 0;

                                    foreach($c_method_text_temp as $c_method_temp_piece){
```

```
                        if(strpos(strtolower($c_method_temp_piece), "(") !== false &&
$c_method_count_match == 0){

                                $c_method_count_match = $c_method_text_count;

                                #echo $method_temp_piece;

                        }

                        $c_method_text_count++;

                }

                $c_method_name_temp = $c_method_text_temp[$c_method_count_match];
                $c_method_name_temp = explode('(', $c_method_name_temp);
                $method_name = $c_method_name_temp[0];

                $c_method_parameter_temp = explode(')', $c_method_name_temp[1]);
                $method_parameter = $c_method_parameter_temp[0];

                #echo $method_parameter;



                $method_return_type = $package_name."."$method_name;
                #echo $method_return_type;

                if(strpos($method_text, " throws ") !== false){

                        $c_throws_pos = strpos($method_text, " throws ");
```

```
                                        if($c_throws_pos > 0){

                                                $c_method_throws_temp = substr($method_text,
$c_throws_pos, strlen($method_text) - $c_throws_pos);

                                                $c_method_throws_temp = str_replace(' throws ', '',
$c_method_throws_temp);

                                                $method_exception = str_replace(';', '',
$c_method_throws_temp);

                                                #$method_exception = str_replace(';', '',
$c_method_throws_temp[sizeof($c_method_throws_temp)-1]);

                                                #echo $method_exception;

                                        }

                                }

                                $final_method_array = array($package_name, $class_type, $class_name, $class_abstract,
$class_static, $class_final, $class_deprecated, $class_visibility, $class_extends, $class_implements, $method_abstract,
$method_synchronized, $method_exception, $method_return_type, $method_name, $method_transient, $method_volatile,
$method_value, $method_parameter, $method_static, $method_final, $method_deprecated, $method_visibility, $method_type, $url);

                                if(!($method_type == 'enumerator value')){

                                        array_push($final_method_table, $final_method_array);
```

```
                    }

                }

                #$class_type =

                #echo $class;

                #var_dump($class);

                #

            }

            $this_package = array();

        }

        /*

        foreach($print_table as $print_this_row){
```

```
            fputcsv($fp, $print_this_row);

}

*/

#$url_array = array();

/*array_push($url_array, 'C:\android-source-doc\3');

#print_r($result_table);

fclose($fp);
```

```
        $finish_file = fopen("finish.txt", "w");
        fwrite($finish_file, "done.");
        fclose($finish_file);

        */

        return array($final_package_table, $final_class_table, $final_method_table);

}


$local_dir = 'C:\Users\azvorinji\Documents\android api\\';

$files = scandir($local_dir);

$url_list = array();

foreach($files as $file){

        if(strpos($file, ".txt") !== false){

                array_push($url_list, $local_dir.$file);

        }

}

#$scrape_file = 'C:\Users\azvorinji\Documents\android api\4.0.1_r1.txt';
#$scrape_file = 'C:\Users\azvorinji\Documents\android api\6.0.0_r1.txt';
```

```
#$text_file_content = file_get_contents($scrape_file);

$package_header_array = array("package_name", "version");

$package_scrape_file = 'package_3.csv';

if(!file_exists($package_scrape_file)){

        $fp = fopen($package_scrape_file, 'a');

        fputcsv($fp, $package_header_array);

}else{

        $fp = fopen($package_scrape_file, 'a');

}

$class_header_array = array("package_name", "class_type", "class_name", "class_abstract", "class_static", "class_final",
"class_deprecated", "class_visibility", "class_extends", "class_implements", "version");

$class_scrape_file = 'class_3.csv';

if(!file_exists($class_scrape_file)){

        $fc = fopen($class_scrape_file, 'a');
```

```
        fputcsv($fc, $class_header_array);

}else{

        $fc = fopen($class_scrape_file, 'a');

}

$method_header_array = array("package_name", "class_type", "class_name", "class_abstract", "class_static", "class_final",
"class_deprecated", "class_visibility", "class_extends", "class_implements", "method_abstract", "method_synchronized",
"method_exception", "method_return_type", "method_name", "method_transient", "method_volatile", "method_value",
"method_parameter", "method_static", "method_final", "method_deprecated", "method_visibility", "method_type", "version");

$method_scrape_file = 'method_3_14.csv';

if(!file_exists($method_scrape_file)){

        $fm = fopen($method_scrape_file, 'a');

        fputcsv($fm, $method_header_array);

}else{

        $fm = fopen($method_scrape_file, 'a');

}
```

```
foreach($url_list as $url){


        $return_table = parse($url);


        $package_table = $return_table[0];
        $class_table = $return_table[1];
        $method_table = $return_table[2];


        $print_table_package = array();
        $print_table_class = array();
        $print_table_method = array();


        $dump = '';


        foreach($package_table as $package_array){


                $print_array_package = array();


                foreach($package_array as $package_value){


                        $dump = $package_value;


                        $dump = trim(preg_replace('/\s\s+/', ' ', $dump));
                        $dump = str_replace("n\\", '', $dump);
```

```
                $dump = str_replace("r\\", '', $dump);

                $dump = str_replace("t\\", '', $dump);

                $dump = str_replace("n\r\\", '', $dump);

                $dump = str_replace("r\n\\", '', $dump);

                $dump = preg_replace('~[\r\n]+~', '', $dump);


                $package_value = $dump;


                array_push($print_array_package, $package_value);


        }


        array_push($print_table_package, $print_array_package);


}


foreach($class_table as $class_array){


        $print_array_class = array();


        foreach($class_array as $class_value){


                $dump = $class_value;

                $dump = trim(preg_replace('/\s\s+/', ' ', $dump));

                $dump = str_replace("n\\", '', $dump);

                $dump = str_replace("r\\", '', $dump);

                $dump = str_replace("t\\", '', $dump);

                $dump = str_replace("n\r\\", '', $dump);

                $dump = str_replace("r\n\\", '', $dump);
```

```
                $dump = preg_replace('~[\r\n]+~', '', $dump);

                $class_value = $dump;

                array_push($print_array_class, $class_value);

        }

        array_push($print_table_class, $print_array_class);

}

foreach($method_table as $method_array){

        $print_array_method = array();

        foreach($method_array as $method_value){

                $dump = $method_value;
                $dump = trim(preg_replace('/\s\s+/', ' ', $dump));
                $dump = str_replace("n\\", '', $dump);
                $dump = str_replace("r\\", '', $dump);
                $dump = str_replace("t\\", '', $dump);
                $dump = str_replace("n\r\\", '', $dump);
                $dump = str_replace("r\n\\", '', $dump);
                $dump = preg_replace('~[\r\n]+~', '', $dump);

                $method_value = $dump;
```

```
                array_push($print_array_method, $method_value);

            }

            array_push($print_table_method, $print_array_method);

        }

        foreach($print_table_package as $print_this_row){

            fputcsv($fp, $print_this_row);

        }

        foreach($print_table_class as $print_this_row){

            fputcsv($fc, $print_this_row);

        }

        foreach($print_table_method as $print_this_row){

            fputcsv($fm, $print_this_row);

        }


}
```

```
$finish_file = fopen("finish.txt", "w");

fwrite($finish_file, "done.");

fclose($finish_file);


?>

</pre>
```

R Script to Re-Structure Data from API Levels 1-13:

```
#install related packages

install.packages("plyr")

install.packages("tidyr")

library(plyr)


#pull in and clean the data files

setwd("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/3.2.4_r1")


package <- read.table("package.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

colnames(package)[1] <- "package_Id"

colnames(package)[2] <- "package_name"


class <- read.table("class.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

colnames(class)[1] <- "class_id"

class$class_id <- paste("class", class$class_id)

colnames(class)[2] <- "class_name"

colnames(class)[3] <- "class_extends"

colnames(class)[4] <- "class_abstract"

colnames(class)[5] <- "class_static"
```

```
colnames(class)[6] <- "class_final"

colnames(class)[7] <- "class_deprecated"

colnames(class)[8] <- "class_visibility"

class$class_type <- "class"

class <- merge(x = package, y = class, by = "package_Id", all.y = TRUE)


interface <- read.table("interface.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

colnames(interface)[1] <- "class_id"

interface$class_id <- paste("interface", interface$class_id)

colnames(interface)[2] <- "class_name"

colnames(interface)[3] <- "class_abstract"

colnames(interface)[4] <- "class_static"

colnames(interface)[5] <- "class_final"

colnames(interface)[6] <- "class_deprecated"

colnames(interface)[7] <- "class_visibility"

interface$class_type <- "interface"

interface$class_extends <- ""

interface <- interface[,c(1,2,10,3,4,5,6,7,8,9)]

interface <- merge(x = package, y = interface, by = "package_Id", all.y = TRUE)


#combine class and interface files

final_class <- rbind(class,interface)


# join implements with class


implements <- read.table("implements.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

colnames(implements)[1] <- "class_implements"

implements$class_Id <- ifelse(is.na(implements$class_Id),"", paste("class",implements$class_Id))

implements$interface_Id <- ifelse(is.na(implements$interface_Id),"", paste("interface",implements$interface_Id))
```

```
implements$class_Inter_id <- paste(implements$class_Id,implements$interface_Id)

trim <- function (x) gsub("^\\s+|\\s+$", "", x)

implements$class_Inter_id <- trim(implements$class_Inter_id)

implements$class_Id <- NULL

implements$interface_Id <- NULL

colnames(implements)[2] <- "class_id"

implements <- aggregate(class_implements~class_id, paste,collapse = ",", data = implements)


final_class <- merge(x = final_class, y = implements, by = "class_id", all = TRUE)

final_class <- final_class[,c(2,1,3,11,4,6,7,8,9,10,5,12)]

final_class$version <- "3.2.4"


# pull in method, constructor and field datasets


construct <- read.table("constructor.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

colnames(construct)[1] <- "Id"

construct$Id <- paste("construct",construct$Id)

colnames(construct)[8] <- "class_Inter_id"

construct$class_Inter_id <- paste("class",construct$class_Inter_id)

construct$transient <- NA

construct$volatile <- NA

construct$value <- NA

construct$return <- NA

construct$abstract <- NA

construct$native <- NA

construct$synchronized <- NA

construct <- construct[,c(1,2,3,9,10,11,12,13,14,15,4,5,6,7,8)]

construct$category <- "constructor"
```

```
field <- read.table("field.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

field$class_Id <- ifelse(is.na(field$class_Id),"", paste("class",field$class_Id))

field$interface_Id <- ifelse(is.na(field$interface_Id),"", paste("interface",field$interface_Id))

field$class_Inter_id <- paste(field$class_Id,field$interface_Id)

field$class_Id <- NULL

field$interface_Id <- NULL

field$Id <- field$Id <- seq(1:nrow(field))

field <- field[,c(11,1,2,3,4,5,6,7,8,9,10)]

field$Id <- paste("field",field$Id)

field$return <- NA

field$abstract <- NA

field$native <- NA

field$synchronized <- NA

field <- field[,c(1,2,3,4,5,6,12,13,14,15,7,8,9,10,11)]

colnames(field)[2] <- "name"

trim <- function (x) gsub("^\\s+|\\s+$", "", x)

field$class_Inter_id <- trim(field$class_Inter_id)

field$category <- "field"


method <- read.table("method.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

colnames(method)[1] <- "Id"

method$Id <- paste("method",method$Id)

method$class_Id <- ifelse(is.na(method$class_Id),"", paste("class",method$class_Id))

method$interface_Id <- ifelse(is.na(method$interface_Id),"", paste("interface",method$interface_Id))

method$class_Inter_id <- paste(method$class_Id,method$interface_Id)

method$class_Id <- NULL

method$interface_Id <- NULL

method$type <- NA

method$transient <- NA
```

```
method$volatile <- NA

method$value <- NA

method <- method[,c(1,2,12,13,14,15,3,4,5,6,7,8,9,10,11)]

trim <- function (x) gsub("^\\s+|\\s+$", "", x)

method$class_Inter_id <- trim(method$class_Inter_id)

method$category <- "method"



#combine constructor, filed and method

method_all <- rbind(method, field, construct)

colnames(method_all) <-
c("method_id","method_name","method_type","method_transient","method_volatile","method_value","method_return","method_abstract","method_native","method_synchronized","method_static","method_final","method_deprecated","method_visibility","class_id","method_category")



#join method_all with exception and parameter

exception <- read.table("exception.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

exception$method_Id <- ifelse(is.na(exception$method_Id),"", paste("method",exception$method_Id))

exception$constructor_Id <- ifelse(is.na(exception$constructor_Id),"", paste("construct",exception$constructor_Id))

exception$Id <- paste(exception$method_Id,exception$constructor_Id)

exception$method_Id <- NULL

exception$constructor_Id <- NULL

trim <- function (x) gsub("^\\s+|\\s+$", "", x)

exception$Id <- trim(exception$Id)

colnames(exception)[1] <- "name"

exception$type_name <- paste(exception$type," ",exception$name)

exception$name <- NULL

exception1 <- aggregate(type~Id, paste,collapse = ",", data = exception)

exception2 <- aggregate(type_name~Id, paste,collapse = ",", data = exception)

exception <- cbind(exception1,exception2)

exception <- exception[,-1]

colnames(exception)[1] <- "exception_type"
```

```
colnames(exception)[3] <- "exception_type_name"

colnames(exception)[2] <- "method_id"


parameter <- read.table("parameter.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

parameter$method_Id <- ifelse(is.na(parameter$method_Id),"", paste("method",parameter$method_Id))

parameter$constructor_Id <- ifelse(is.na(parameter$constructor_Id),"", paste("construct",parameter$constructor_Id))

parameter$Id <- paste(parameter$method_Id,parameter$constructor_Id)

parameter$method_Id <- NULL

parameter$constructor_Id <- NULL

trim <- function (x) gsub("^\\s+|\\s+$", "", x)

parameter$Id <- trim(parameter$Id)

parameter$type_name <- paste(parameter$type," ",parameter$ï..name)

parameter$ï..name <- NULL

parameter1 <- aggregate(type~Id, paste,collapse = ",", data = parameter)

parameter2 <- aggregate(type_name~Id, paste,collapse = ",", data = parameter)

parameter <- cbind(parameter1,parameter2)

parameter <- parameter[,-3]

colnames(parameter)[2] <- "parameter_type"

colnames(parameter)[3] <- "parameter_type_name"

colnames(parameter)[1] <- "method_id"


method_final <- join(method_all,exception, by = "method_id",type = "full", match = "all")

method_final <- join(method_final, parameter, by = "method_id", type = "full", match = "all")


#merge method_final with final_class

final_method <- join(final_class, method_final, by = "class_id", type = "full", match = "all")

final_method <- final_method[,c(3,4,5,6,7,8,9,10,11,12,21,23,29,16,20,15,17,18,19,32,24,25,26,27,28,13)]


final_method$class_abstract <- ifelse(final_method$class_abstract == "true",1,0)
```

```
final_method$class_static <- ifelse(final_method$class_static == "true",1,0)

final_method$class_final <- ifelse(final_method$class_final == "true",1,0)

final_method$class_deprecated <- ifelse(final_method$class_deprecated == "deprecated",1,0)

final_method[is.na(final_method)] <- ""

final_method$method_abstract <-
ifelse(final_method$method_abstract=="false",0,ifelse(final_method$method_abstract=="true",1,""))

final_method$method_synchronized <-
ifelse(final_method$method_synchronized=="false",0,ifelse(final_method$method_synchronized=="true",1,""))

colnames(final_method)[13] <- "method_exception"

final_method$method_return_type <- paste(final_method$method_type,final_method$method_return)

final_method <- final_method[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,27,16,17,18,19,20,21,22,23,24,25,26)]

final_method$method_transient <-
ifelse(final_method$method_transient=="false",0,ifelse(final_method$method_transient=="true",1,""))

final_method$method_volatile <-
ifelse(final_method$method_volatile=="false",0,ifelse(final_method$method_volatile=="true",1,""))

colnames(final_method)[19] <- "method_parameter"

final_method$method_static <- ifelse(final_method$method_static == "true",1,0)

final_method$method_final <- ifelse(final_method$method_final == "true",1,0)

final_method$method_deprecated <- ifelse(final_method$method_deprecated == "deprecated",1,0)

colnames(final_method)[24] <- "method_type"



final_class <- final_method[,c(1,2,3,4,5,6,7,8,9,10,25)]

package$version <- "3.2.4"

final_package <- package[,c(2,3)]



#save to csv file

write.table(final_package, "C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android
evolution/android api/outcome/final_package3.2.4.csv", sep = ",", row.names= FALSE)

write.table(final_class, "C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android
evolution/android api/outcome/final_class3.2.4.csv", sep = ",", row.names= FALSE)

write.table(final_method, "C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android
evolution/android api/outcome/final_method3.2.4.csv", sep = ",", row.names= FALSE)
```

############ above code changed becasue of method exception in the final data only needs method_type###

#read all_versions

package1.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package1.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class1.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class1.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method1.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method1.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

package1.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package1.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class1.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class1.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method1.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method1.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

package1.5 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package1.5.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class1.5 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class1.5.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method1.5 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method1.5.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

package1.6 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package1.6.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class1.6 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class1.6.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method1.6 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method1.6.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

package2.0.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package2.0.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class2.0.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class2.0.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

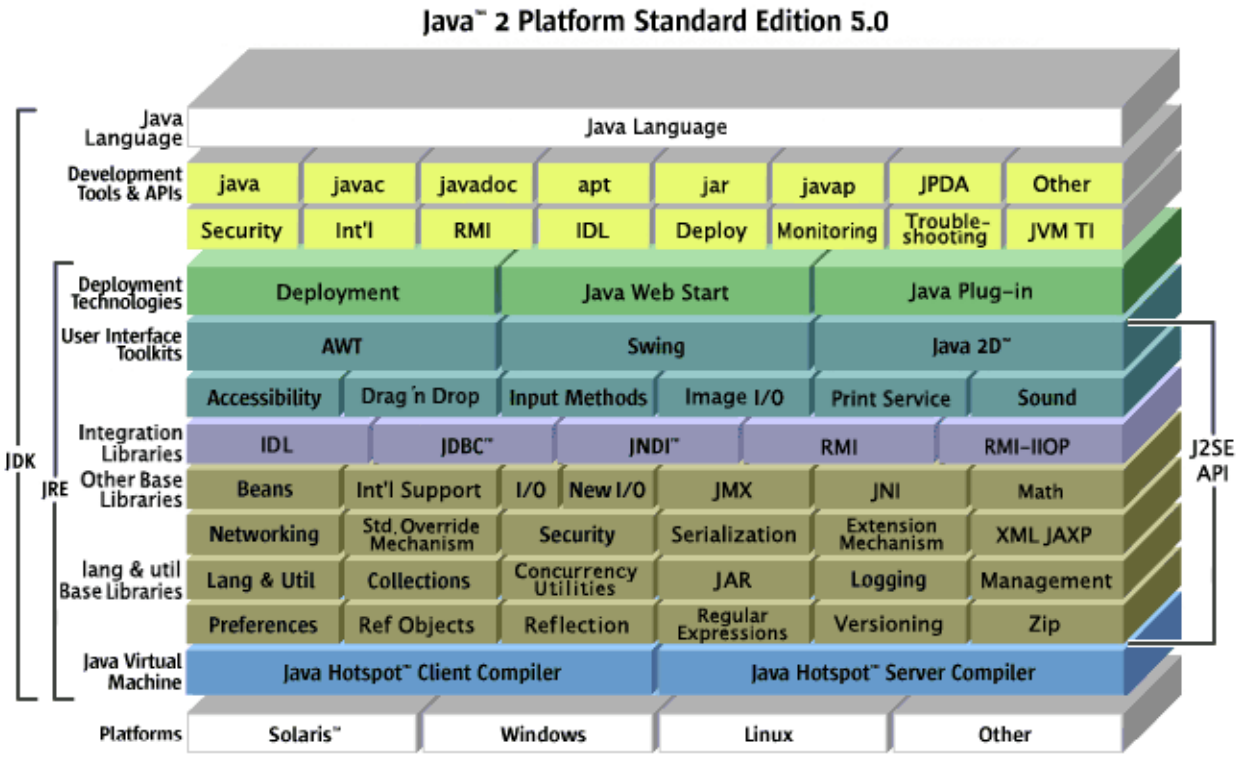
method2.0.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method2.0.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

```
package2.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package2.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class2.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class2.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method2.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method2.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package2.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package2.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class2.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class2.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method2.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method2.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package2.2 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package2.2.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class2.2 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class2.2.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method2.2 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method2.2.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package2.3.3 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package2.3.3.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class2.3.3 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class2.3.3.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method2.3.3 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method2.3.3.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package2.3 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package2.3.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class2.3 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class2.3.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method2.3 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method2.3.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package3.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package3.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class3.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class3.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)
```

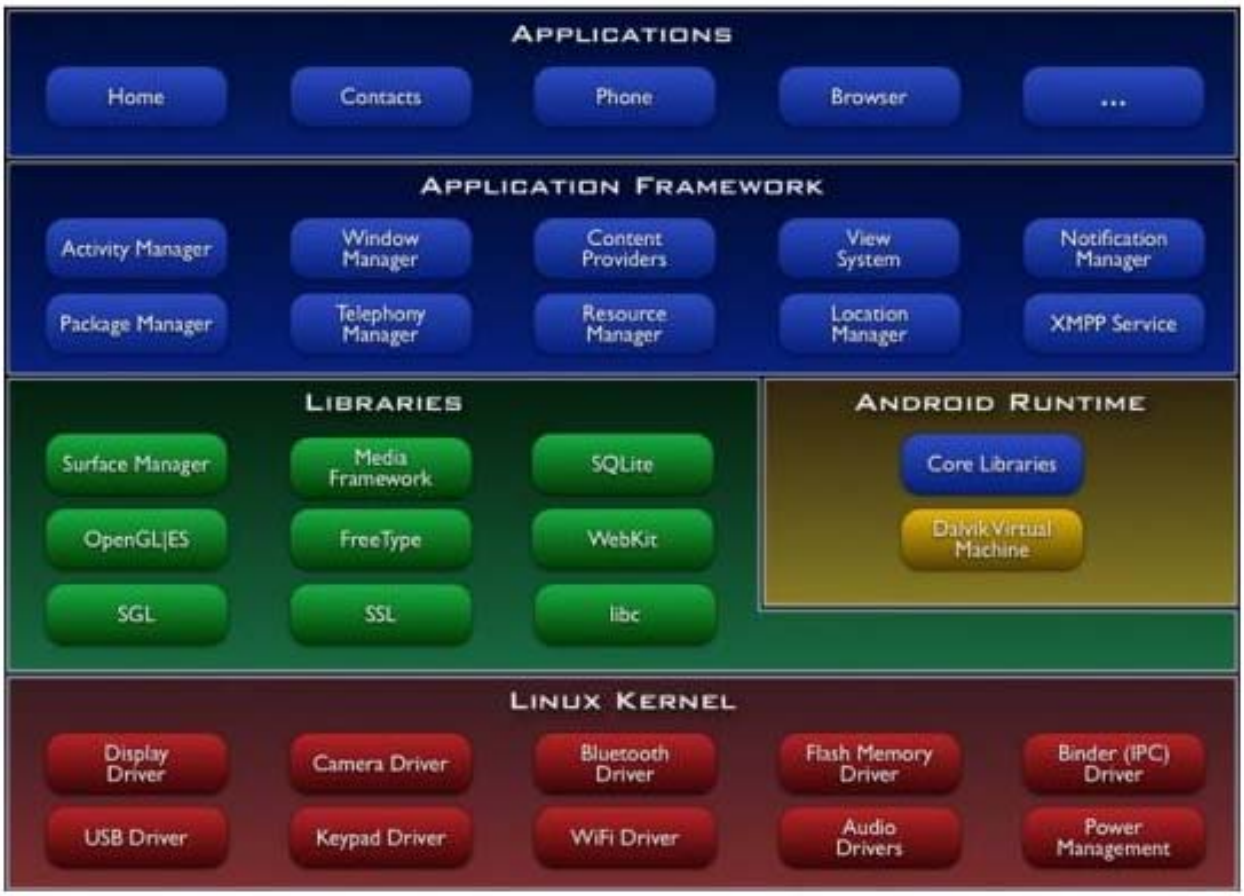
```
method3.0 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method3.0.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package3.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package3.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class3.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class3.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method3.1 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method3.1.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package3.2.4 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_package3.2.4.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class3.2.4 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_class3.2.4.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method3.2.4 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/final_method3.2.4.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


package14_23 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/package14-23.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

class14_23 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/class14-23.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

method14_23 <- read.table("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/method14-23.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


# combine all versions

package_all_version <-
rbind(package1.0,package1.1,package1.5,package1.6,package2.0.1,package2.0,package2.1,package2.2,package2.3.3,package2.3,package3.0,package3.1,package3.2.4,package14_23)

class_all_version <-
rbind(class1.0,class1.1,class1.5,class1.6,class2.0.1,class2.0,class2.1,class2.2,class2.3.3,class2.3,class3.0,class3.1,class3.2.4,class14_23)

method_all_version <-
rbind(method1.0,method1.1,method1.5,method1.6,method2.0.1,method2.0,method2.1,method2.2,method2.3.3,method2.3,method3.0,method3.1,method3.2.4,method14_23)


# save final all version files

write.table(package_all_version, "C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/package_all_version.csv", sep = ",", row.names= FALSE)

write.table(class_all_version, "C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/class_all_version.csv", sep = ",", row.names= FALSE)
```

```
write.table(method_all_version, "C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api/outcome/method_all_version.csv", sep = ",", row.names= FALSE)


method1_2 <- merge(x = method1.0,y = method1.1, by = c("package_name","class_name","method_name"),all.x = TRUE)


setwd("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/22_API popularity analysis/android evolution/android api 1-23 clean")


method_all_version <- read.table("method_all_version.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)


method4.0 <- subset(method_all_version,version == "4.0.1")

method4.2 <- subset(method_all_version,version == "4.2")

method4.4 <- subset(method_all_version,version == "4.4")

method5.0 <- subset(method_all_version,version == "5")
```

APPENDIX J – Java SE 5.0 Platform diagram



APPENDIX K – Android Platform



APPENDIX L – Android Version Names

(http://developer.android.com/guide/topics/manifest/uses-sdk-element.html)

API Level	Version
23	Marshmallow
22	Lollipop
21	
20	KitKat
19	
18	Jelly Bean
17	
16	
15	Ice Cream Sandwich
14	
13	Honeycomb
12	
11	
10	Gingerbread
9	
8	Froyo
7	Eclair
6	
5	
4	Donut
3	Cupcake
2	Base
1	

Platform Version	API Level	VERSION_CODE
Android 6.0	23	M
Android 5.1	22	LOLLIPOP_MR1
Android 5.0	21	LOLLIPOP

Android 4.4W	20	KITKAT_WATCH
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

APPENDIX M – PageRank Data

PageRank scores for the copied classes in the Android source code, averaged by the package to which each class belongs.

Package Name	PageRank	Package Name	PageRank
java.lang	0.005419	java.util.logging	1.48E-05
java.lang.annotation	0.004816	java.security.interfaces	1.35E-05
java.io	0.000739	javax.security.auth.x500	3.56E-05
java.nio.charset	0.000738	javax.crypto	1.32E-05
java.util	0.000409	java.util.zip	1.5E-05
java.util.regex	0.002151	java.util.prefs	1.21E-05
java.nio	0.00033	javax.crypto.spec	1.18E-05
java.lang.reflect	0.000564	javax.security.cert	1.32E-05
java.security	0.000115	java.util.jar	1.23E-05
java.lang.ref	0.000472	javax.security.auth.callback	1.36E-05
java.net	3.83E-05	javax.security.auth	1.09E-05
java.text	3.76E-05	javax.sql	1.19E-05
java.nio.channels.spi	0.000113	java.beans	9.98E-06
java.nio.channels	3.27E-05	javax.net	1.1E-05
java.sql	1.61E-05	java.security.acl	1.18E-05
java.nio.charset.spi	0.000147	javax.crypto.interfaces	1.04E-05
java.security.cert	1.88E-05	javax.security.auth.login	9.27E-06
java.security.spec	1.69E-05	java.awt.font	8.42E-06
javax.net.ssl	1.59E-05		

APPENDIX N – Timeline of Facts

Date	Fact
6/17/1995	The Java programming language and the Java platform--a software platform that includes a runtime environment to execute programs written in the Java programming language--were developed by Sun Microsystems. The Java platform is first released in 1995.
9/1/2003	Android organization incorporates.
Year 2005	"In 2005, Java had evolved to primarily be used by mobile carriers . . . Sun's strategy was to have mobile devices, which were well suited to Java." -Eric Schmidt on the witness stand in 2012.
8/17/2005	Google acquires Android.
3/16/2006	Vineet Gupta (Sun) emails Andy Rubin kicking off discussions about a license.
Approx. May 2006	Google decides to go forward with Java APIs without a license.
Years 2006 and 2007	Sun released some of the source code for Java SE and other editions of the Java Platform under the terms of the GNU Public License, version 2 (GPLv2) open source license.
6/1/2007	Apple's iPhone is released in the US.
11/5/2007	Google announces Android and the Open Handset Alliance.
8/23/2008	Android 1.0 (API level 1), the first commercial version of the software, is released.
10/22/2008	The first commercial smartphone running Android, the HTC Dream, was released. The Android Marketplace, predecessor to the Google Play store, is announced.
2/9/2009	Android 1.1 update (API level 2) is released
4/27/2009	Android 1.5 update "Cupcake" (API level 3) is released.
9/15/2009	Android 1.6 SDK "Donut" (API level 4) is released.
10/26/2009	Android 2.0 SDK "Éclair" (API level 5) is released.
12/3/2009	Android 2.0.1 "Éclair" (API level 6) is released.
1/1/2010	Oracle's acquisition of Sun Microsystems, which is renamed Oracle America, Inc., is complete. Oracle acquires Sun for $7.4 billion.
1/5/2010	The Google Nexus One, manufactured by HTC, is launched. Google Nexus products run Android and are designed, developed, and supported by Google and partly developed and wholly manufactured by OEMs.
1/12/2010	Android 2.0.1 "Éclair" (API level 7) is released.
5/20/2010 - 11/21/2011	Android 2.2 - 2.2.3 SDK "Froyo" (API level 8) is released.
Summer 2011	[redacted]
10/1/2010	"To date, the Java platform has attracted more than 6.5 million software developers."
10/27/2010	Oracle filed Amended Complaint in N.D.Cal. for alleged copyright infringement.
11/1/2010	In Google's Answer to Oracle's Amended Complaint, Google states that there are

Date	Fact
	approximately 200,000 Android-based handsets activated every day on over fifty different wireless carriers worldwide. There are approximately 90 different Android-based devices made by over 20 different manufacturers. Android Market, predecessor to the Google Play store, has over 80,000 apps available for download.
12/6/2010	The second Nexus product, Google Nexus S, is launched. Manufactured by Samsung, it is the first phone running Android 2.3 Gingerbread.
Winter 2010	[redacted].
12/6/2010 - 1/31/2011	Android 2.3 - 2.3.2 SDK "Gingerbread" (API level 9) is released.
2/9/2011 - 9/21/2011	Android 2.3.3 - 2.3.7 SDK "Gingerbread" (API level 10) is released.
2/22//2011	Android 3.0 SDK "Honeycomb" (API level 11) is released.
5/10/2011	Android 3.1 "Honeycomb" (API level 12) is released.
7/15/2011 - 2/29/2012	Android 3.2 - 3.2.6 SDK "Honeycomb" (API level 13) is released. The first- and second-generation Google TV-enabled devices utilize Honeycomb 3.2.
10/18/2011 - 11/28/2011	Android 4.0 - 4.0.2 SDK "Ice Cream Sandwich" (API level 14) is released.
11/17/2011	The third Nexus product, Galaxy Nexus, is launched. Manufactured by Samsung, its release coincides with the launch of Android 4.0 Ice Cream Sandwich.
12/16/2011 - 3/29/2012	Android 4.0.3 - 4.0.4 "Ice Cream Sandwich" (API level 15) is released.
4/16/2012	Copyright portion of trial began.
5/7/2012	Patent portion of trial began.
7/9/2012 - 10/9/2012	Android 4.1 - 4.1.2 "Jelly Bean" (API level 16) is released.
7/9/2012	Android 4.1 Jelly Bean was released to the Android Open Source Project on July 9, 2012.
11/13/2012	The fourth Nexus product, Google Nexus 4 (LG Mako), is launched. Manufactured by LG, its release coincides with the launch of Android 4.2 Jelly Bean.
11/13/2012 - 2/11/2013	Android 4.2 - 4.2.2 "Jelly Bean" (API level 17) is released.
7/24/2013 - 10/3/2013	Android 4.3 - 4.3.1 "Jelly Bean" (API level 18) is released.
10/31/2013	The fifth Nexus product, Google Nexus 5, is launched. Manufactured by LG, it is the first phone to run Android 4.4 KitKat.
10/31/2013 - 6/19/2014	Android 4.4 - 4.4.4 "KitKat" (API level 19) is released.
5/9/2014	Federal Circuit decision released. The court finds that the structure, sequence, and organization of Java APIs are copyrightable and remands the case to be tried on fair use.
6/25/2014 - 10/21/2014	Android 4.4W - 4.4W.2 "KitKat" (API level 20) is released. Same as Android 4.4 but with wearable extensions added.

Date	Fact
6/25/2014	Initial release of Android Wear platform for smartwatches.
10/15/2014	The sixth Nexus product, Google Nexus 6, is launched. Manufactured by Motorola Mobility, it runs Android 5.0 Lollipop.
11/12/2014 - 12/19/2014	Android 5.0 - 5.0.2 "Lollipop" is released.
3/9/2015 - 4/21/2015	Android 5.1 - 5.1.1 "Lollipop" is released.

APPENDIX O – "Brillo" Documentation

226. Within the Brillo source code there was a directory called /brillo/prebuilt/sdk. This directory is not documented, but based on my review of the contents, it appears to contain the .class files for the Android API. The directory contains 25 subdirectories named 1 through 23, current, and system current. See GOOG-SC10000001-GOOG-SC10002234. These appear to correspond to the different versions of Android. For example, Android Marshmallow—the current version commercially available—corresponds to API Level 23.

227. Within each of these 25 subdirectories was a file called android.jar. A .jar file is a Java Archive, a file format that allows one to bundle multiple files together in a single archive.[121] A .jar file is packaged with the.zip file format and can be unpackaged the way any .zip file can.

228. For each of the 25 different versions of android.jar in the various subdirectories of /brillo/prebuilt/sdk, the file was unpackaged, creating a set of directories and files. The directories created were:

/android/
/com/
/dalvik/
/java/
/javax/
/junit/
/org/

229. These directories correspond to the top-level namespaces in the Android API. See http://developer.android.com/reference/packages.html.

230. A listing of all the directories and files contained in the 25 versions of android.jar was created by performing a directory listing on those files and directories created when the 25 versions of android.jar were unpackaged and saving the output of that directory listing to a file named brillo_prebuilt_sdk_jar.txt. I understand Google produced this file as GOOG-SC10000001.

231. I analyzed the contents of GOOG-SC10000001, paying particular attention to the \java\ and \javax\ directories, which is where I would expect the 37 Java API Packages to be found.

[121] https://docs.oracle.com/javase/tutorial/deployment/jar/index.html

232. For each version of Android, I found directories that correspond to the package structure of the 37 Java API Packages. For example, in the folder corresponding to API Level 9, I found the following directories:

/9/java/awt/font/
/9/java/beans/
/9/java/io/
/9/java/lang/
/9/java/lang/annotation/
/9/java/lang/ref/
/9/java/lang/reflect/
/9/java/net/
/9/java/nio/
/9/java/nio/channels/
/9/java/nio/channels/spi/
/9/java/nio/charset/
/9/java/nio/charset/spi/
/9/java/security/
/9/java/security/acl/
/9/java/security/cert/
/9/java/security/interfaces/
/9/java/security/spec/
/9/java/sql/
/9/java/text/
/9/java/util/
/9/java/util/jar/
/9/java/util/logging/
/9/java/util/prefs/
/9/java/util/regex/
/9/java/util/zip/
/9/javax/crypto/
/9/javax/crypto/interfaces/
/9/javax/crypto/spec/
/9/javax/net/
/9/javax/net/ssl/
/9/javax/security/auth/
/9/javax/security/auth/callback/
/9/javax/security/auth/login/
/9/javax/security/auth/x500/
/9/javax/security/cert/
/9/javax/sql/

233. This directory structure reflects the sequence, structure, and organization ("SSO") of the 37 Java API Packages a jury already determined that Google improperly copied from Oracle. It also reflects the SSO that Mr. Zeidman concludes was copied in his report. (See Zeidman Report at ¶¶112-119)

234. Furthermore, the .class files contained in these directories appear to be the same files that Mr. Zeidman concluded contain Oracle's declaring code. For example, in the folder for API Level 9, there is a folder that corresponds to the java.awt.font package (/9/java/awt/font/). In this folder there are two .class

files: NumericShaper.class and TextAttribute.class. Mr. Zeidman concluded, based on his source code analysis, that declaring code for these two classes is present in Android API Level 9.[122] The .class files I see listed here appear to be compiled versions of the files Mr. Zeidman found contain Oracle's declaring code.

235. Along with my research assistant, I made a similar review of the directory structure and .class files for each version of android.jar we encountered.

[122] <Insert Cite.>

APPENDIX P – List of copied API packages

Copied Oracle API	API Purpose	First Android Version to Copy
java.awt.font	AWT (Abstract Window Toolkit) provides a set of native user interface components (such as the display of graphics and fonts), and an event-handling model (e.g. to handle user clicks in the user interface). It includes graphics and imaging tools to display shapes, and color, and to "render" fonts, i.e. to display text and other typographic features in a visible way (e.g. to display text on the screen with line breaks at appropriate places in the group of characters that make up the text). It also includes tools to facilitate flexible window layouts in a flexible way that does not depend on the window size or the screens resolution. It also enables "cut and paste" capabilities for the clipboard.[123] Java.awt.font more speficially supports fonts such as "TrueType" or "PostScript" and facilitates displaying text using the different typographic designs of various fonts, as well as using text attributes like "bold."[124]	1.0 (Level 1 - "Base")
java.beans	Provides code that aids developers in creating reusable software components for applications that use the JavaBeans™ architecture. JavaBeans™ is a portable, platform-independent component model that enables developers to write reusable components once and run them anywhere, and it can act as a bridge to other proprietary component models).[125] JavaBeans are typically deployed in a network or distributed application environment, and they provide ways for components to work together that do not necessarily "know about each other" in advance. This package also provides capabilities such as a *long term* "persistence" approach, i.e. a way to store the data (or state) associated with a component (bean) in a database or other data repository (e.g. using XML format).[126]	1.5 (Level 3 - "Cupcake")

[123] Abstract Window Toolkit (AWT), ORACLE, http://docs.oracle.com/javase/7/docs/technotes/guides/awt/index.html (last visited January 8, 2016).

[124] CLASS FONT, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/awt/Font.html (last visited January 8, 2016).

[125] JAVABEANS FAQ: GENERAL QUESTIONS, ORACLE, http://www.oracle.com/technetwork/java/javase/faq-135947.html (last visited January 8, 2016).

[126] PACKAGE JAVABEANS, ORACLE, http://docs.oracle.com/javase/7/docs/api/java/beans/package-summary.html (last visited January 8, 2016).

Copied Oracle API	API Purpose	First Android Version to Copy
java.io	Provides components for reading and writing data (input and output), such as reading or writing data from or to a file or other source located over a network. The classes of this package can take data representing various types, like characters, numbers or other information; turn them into a sequence of binary bits; stream them; and then reconstruct the appropriate type(s) of information (i.e. characters, numbers, etc...).[127]	1.0 (Level 1 - "Base")
java.lang	The package provides "system operations" that manage the dynamic loading of classes, the creation of new operating system processes or processor threads (to facilitate concurrent processing), the querying of the host environment for information such as the time of day, and the enforcement of security policies. The package also defines wrapper classes and data type conversions for certain data types that may be represented as objects, such as Boolean, Character, Integer, Long, Short, Float and Double, and defines objects, classes and defines state and behaviors such the ability of an object to compare itself to another object, to convert itself to a string, to wait (suspend execution) on the condition of a variable or notification, to notify other objects of the change in a condition variable, and to return the objects class name. In addition classes in this package enable capabilities for other classes such as cloning of an object, and garbage collection (i.e. freeing up the memory associated with an object when there are no longer any references to it).[128]	1.0 (Level 1 - "Base")
java.lang.annotation	Provides a facility for developers to add “meta data” to their code which serves the purpose of providing compiler instructions, build-time instructions, and runtime instructions. In other words, annotations affect the way tools and libraries treat a program; normally the annotations to not affect program execution. As an example, the “@deprecated” annotation can give a compiler instructions that the marked class, method or field has been deprecated and should no longer be used. The compiler will then issue a warning during	1.0 (Level 1 - "Base")

[127] PACKAGE JAVA.IO DESCRIPTION, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html#package_description (last accessed January 8, 2016).

[128] PACKAGE JAVA.LANG, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/lang/package-summary.html (last visited January 8, 2016).

Copied Oracle API	API Purpose	First Android Version to Copy
	compilation.[129]	
java.lang.ref	Provides component that enables interaction with the garbage collector. The package is used to create a wrapper around a “referent”, i.e. an object that the reference points to, that allows the program to remain aware of the garbage collection status of an object without affecting the garbage collection process.[130]	1.0 (Level 1 - "Base")
java.lang.reflect	Provides component to obtain information programmatically about fields, methods and constructors of loaded classes, as well as to use this information to operate on underlying fields, methods and constructors. Classes in this package are often used by applications such as debuggers, interpreters, object inspectors, class browsers, and other services that need access to either the public members of a target object (based on its runtime class) or the members declared by a given class. For example, the classes in this package can be used to determine the length of an array represented by an array object.[131], [132]	1.0 (Level 1 - "Base")
java.net	Provides components for implementing networking applications, roughly divided into two sections: i) ability to deal with addresses like IP addresses, sockets (bi-directional communication mechanisms), and network interfaces; and ii) capabilities to deal with Universal Resource Identifiers, URLs, which represent Universal Resource Locators, and “connections” to the resource pointed to by URLs.[133] As one example, these classes include mechanisms for requesting and performing password authentication over a network.	1.0 (Level 1 - "Base")

[129] ANNOTATIONS, ORACLE, http://docs.oracle.com/javase/7/docs/technotes/guides/language/annotations.html (last visited January 8, 2016).

[130] PACKAGE JAVA.LANG.REF, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/lang/ref/package-summary.html (last visited January 8, 2016).

[131] PACKAGE JAVA.LANG.REFLECT, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/lang/reflect/package-summary.html (last visited January 8, 2016).

[132] JAVA REFLECTION API, ORACLE, http://docs.oracle.com/javase/7/docs/technotes/guides/reflection/index.html (last visited January 8, 2016).

[133] PACKAGE JAVA.NET, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/net/package-summary.html (last visited January 8, 2016).

Copied Oracle API	API Purpose	First Android Version to Copy
java.nio	Provides components for system input and output through data streams, serialization and the file system. Its capabilities are used for data intensive Input/Output operations as an extension to the capabilities provided by the java.io package.[134]	1.0 (Level 1 - "Base")
java.nio.channels	Provides components to define channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets, and defines selectors, for multiplexed, non-blocking I/O operations. Non-blocking means threads of computation do not have to wait for I/O operations to complete before continuing, while multiplexing means multiple connections can be used simultaneously using a "selector."[135]	1.0 (Level 1 - "Base")
java.nio.channels.spi	Provides service-provider components for the java.nio.channels package that is intended to be used by developers who are defining new selector providers or asynchronous channel providers for IO purposes.[136]	1.0 (Level 1 - "Base")
java.nio.charset	Provides components that enable mapping between sequences of sixteen bit Unicode characters and sequences of bytes representing characters. It includes capabilities for retrieving names associated with character sets, decoders to transform bytes into characters of a specific character set, and encoders to transform characters into bytes.[137]	1.0 (Level 1 - "Base")
java.nio.charset.spi	Defines "service-provider" components that would be used by developers who are creating new character sets. The service providers created would then be accessed by other developers through the java.nio.charset package.[138]	1.0 (Level 1 - "Base")
java.security	Package implements the Java security framework for providing fine-grained access control to system resources; for generation and storage of	1.0 (Level 1 - "Base")

[134] JAVA PLATFORM, STANDARD EDITION 7 API SPECIFICATION, http://docs.oracle.com/javase/7/docs/api/index.html (last visited January 8, 2016).

[135] PACKAGE JAVA.NIO.CHANNELS, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/nio/channels/package-summary.html (last visited January 8, 2016).

[136] PACKAGE JAVA.NIO.CHANNELS.SPI, ORACLE, http://docs.oracle.com/javase/7/docs/api/java/nio/channels/spi/package-summary.html (last visited January 8, 2016).

[137] CLASS CHARSET, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/nio/charset/Charset.html (last visited January 8, 2016).

[138] PACKAGE JAVA.NIO.CHARSET.SPI, ORACLE, http://docs.oracle.com/javase/7/docs/api/java/nio/charset/spi/package-summary.html (last visited January 8, 2016).

Copied Oracle API	API Purpose	First Android Version to Copy
	cryptographic public key pairs; for supporting cryptographic operations like message digests and signature generation; and for supporting signed/guarded objects and secure random number generation. At a high level, the package implements capabilities such as providing access control to prevent untrusted code from performing sensitive operations and to provide authentication.[139]	
java.security.acl*	Provides an *interface* to represent an Access Control List data structure used to guard access to resources. The list is a data structure with multiple access control entries that grant or deny access of the associated principal to underlying resources.[140]	1.0 (Level 1 - "Base")
java.security.cert	Provides components for parsing and managing certificates, certificate revocation lists (CRLs), and certification paths. Provides classes for working with identity certificates and certificate revocation lists (CRLs). It defines generic Certificate and CRL classes and X509Certificateand X509CRL classes that provide full support for standard X.509 certificates and CRLs. These classes include a simplified version of the java.security.cert package. A certificate is an object that contains the name of an entity and a public key for that entity.[141]	1.0 (Level 1 - "Base")
java.security.interfaces	Provides components to implement cryptographic algorithms. These components define methods that provide algorithm-specific information, such as key values and parameter values, about public and private keys.	1.0 (Level 1 - "Base")
java.security.spec	Provides components that specify public and private keys and encodings of those keys, a set of parameters used with digital signature algorithms, how a key can be specified.	1.0 (Level 1 - "Base")
java.sql	Provides components for accessing and processing data stored in a data source (usually a relational database). It includes a framework whereby different data source drivers can be installed dynamically to access different data sources. The package is mainly geared to passing SQL statements to a database, but it also provides for	1.0 (Level 1 - "Base")

[139] PACKAGE JAVA.SECURITY, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/security/package-summary.html (last visited January 8, 2016).

[140] PAKCAGE JAVA.SECURITY.ACL, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/security/acl/Acl.html (last visited January 8, 2016).

[141] PACKAGE JAVA.SECURITY.CERT, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/security/cert/package-summary.html#package_description (last visited January 8, 2016).

Copied Oracle API	API Purpose	First Android Version to Copy
	reading and writing data from any data source with a tabular format. The reader/writer facility, (available through the javax.sql.RowSet), can be customized to use and update data from a spread sheet, flat file, or other tabular data sources.[142]	
java.text	Provides components for handling text, dates, numbers, and messages in a manner independent of natural languages. This allows developers to write applications that are language-independent, allowing them to rely on separate, dynamically-linked resources for "localization." This allows developers to add support for additional languages at any time.[143]	1.0 (Level 1 - "Base")
java.util	Provides for the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (e.g. a string tokenizer, a random-number generator, and a bit array).[144] It provides the Enumeration interface for series of items (e.g. days of the week). It provides facilities for manipulating time and date information. Importantly, it provides the collections framework (library) for dealing with lists, stacks, sets and other collections.	1.0 (Level 1 - "Base")
java.util.jar	Provides components for reading and writing the JAR (Java ARchive) file format, which is an archive file whose first entry is a specially named manifest file that contains attributes and digital signatures for the file entries that follow it.	1.0 (Level 1 - "Base")
java.util.logging	Provides components related to logging. Logging is the process of writing log (A Log is record of events or information that can be tracked for future use) messages during the execution of a program to a central place. This logging allows one to report and persist so that they can later be retrieved and analyzed. This package includes: 1. Logger: The object which performs the logging in applications. 2. Level: The log levels define the severity of a message and which messages should be written to the log. 3. Handler: Provides a way of capturing/writing log	1.0 (Level 1 - "Base")

[142] PACKAGE JAVA.SQL, ORACLE https://docs.oracle.com/javase/7/docs/api/java/sql/package-summary.html (last visited January 8, 2016).

[143] PACKAGE JAVA.TEXT, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/text/package-summary.html (last visited January 8, 2016).

[144] PACKAGE JAVA.UTIL, ORACLE, https://docs.oracle.com/javase/7/docs/api/java/util/package-summary.html (last visited January 8, 2016).

Copied Oracle API	API Purpose	First Android Version to Copy
	messages to the file or on console. 4. Formatter: Specifics format for log messages to be written.	
java.util.prefs	Provides components to store and retrieve user and system preference/settings and configuration data, including user preferences and system preferences.	1.0 (Level 1 - "Base")
java.util.regex	Provides components to match or find other strings or sets of strings, using a specialized syntax held in a search pattern. The search pattern can be anything from a simple character, a fixed string or a complex expression containing special characters describing the pattern.	1.0 (Level 1 - "Base")
java.util.zip	Provides components for data compression and decompression, and for reading and writing compressed files.	1.0 (Level 1 - "Base")
javax.crypto	Provides components for cryptographic operations, including encryption, key generation and key agreement, and Message Authentication Code (MAC) generation. It supports symmetric, asymmetric, block, and stream cipher encryption, as well as secure streams and sealed objects.[145]	1.0 (Level 1 - "Base")
javax.crypto.interfaces	Provides components that support public/private key pairs and methods for two computer users to generate a shared private key with which they can then exchange information across an insecure channel. It has interfaces that specify the parameters that generate the key; define a key family and manage actual key values.	1.0 (Level 1 - "Base")
javax.crypto.spec	Provides components that specify public/private keys and encodings of those keys, as well as how a key can be specified. A key may be specified in an algorithm-specific way, or in an algorithm-independent encoding format.	1.0 (Level 1 - "Base")
javax.net	Provides components for networking applications, including for factories for creating sockets. Socket factories allow developers to encapsulate socket creation and configuration behavior.[146] This Factory design provides polymorphism, makes it	1.0 (Level 1 - "Base")

[145] PACKAGE JAVA.CRYPTO, ORACLE, https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html (last visited January 8, 2016).

[146] PACKAGE JAVAX.NET, ORACLE, http://docs.oracle.com/javase/7/docs/api/javax/net/package-summary.html (last visited January 8, 2016).

Copied Oracle API	API Purpose	First Android Version to Copy
	more flexible, so different kinds of sockets can be used by the same application code just by passing it different kinds of factories.	
javax.net.ssl	Provides components that support secure infrastructure for networking applications. Using these classes and interfaces one can communicate reliably into the network and can optionally encrypt the data and/or authenticate the communicating peers.	1.0 (Level 1 - "Base")
javax.security.auth	Provides components for authentication and authorization. It supports various types of authentication modules. The authorization component allows specification of access controls.	1.0 (Level 1 - "Base")
javax.security.auth.callback	Provides components for services to interact with applications in order to retrieve information (authentication data including usernames or passwords, for example) or to display information (error and warning messages, for example).	1.0 (Level 1 - "Base")
javax.security.auth.login	Provides components for configuration related to login and authentication.	1.0 (Level 1 - "Base")
javax.security.auth.x500	Provides components that should be used to store X.500 principals and their credentials. X.500 is a series of protocols for computer networking covering electronic directory services.	1.0 (Level 1 - "Base")
javax.security.cert	Provides components for working with identity certificates and certificate revocation lists (CRLs). A certificate is an object that contains the name of an entity and a public key for that entity. This was an extension for java.security.cert but was added to core J2SE 1.4	1.0 (Level 1 - "Base")
javax.sql	This package supplements the java.sql to provide capabilities for server side access and processing of data from data sources. Similar to the java.sql package it provides ways to make connections to data sources, including relational databases. It includes greater flexibility for making changes to a data source's properties and additional capability for connection and statement "pooling" and distributed database transactions.[147]	1.0 (Level 1 - "Base")

[147] PACKAGE JAVAX.SQL, ORACLE, http://docs.oracle.com/javase/7/docs/api/javax/sql/package-summary.html (last visited January 8, 2016).

APPENDIX Q – R Scripts for Counting of Number of Method Changes

R Script for Android Method Changes:

```
install.packages("dplyr")

library(dplyr)


#pull in the data files

setwd("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/34_API stability analysis/android stability")


android_stability <- read.csv("android stability input.csv", as.is = TRUE)


#group the unique methods in each version and get the unique method list

android_stability <- android_stability %>% group_by(class,package,method,version) %>% mutate(length(m_struct))

names(android_stability)[8] <- "count"


#get the methods with only 1 method strucure in each version

m_unique <- android_stability[android_stability$count==1,]


#get the method with more than 1 method structure in each version

m_multiple <- android_stability[android_stability$count > 1,]

m_multiple$m_str_var <- paste(m_multiple$m_struct,m_multiple$m_var)


write.csv(m_multiple,file='m_multiple.csv')


m_multiple_group <- group_by(m_multiple,class,package,method,version)

m_multiple_updated <- summarize(m_multiple_group,m_struct_c = paste(m_struct, collapse = " "))



#android_stability$m_str_var <- paste(android_stability$m_struct,android_stability$m_var)

#android_stability$m_char <- nchar(android_stability$m_str_var)
```

```
p <- c("android",
       "android.annotation",
       "android.content.res",
       "java.nio.channels",
       "java.nio.channels.spi",
       "android.database",
       "java.nio.charset",
       "java.nio.charset.spi",
       "java.security",
       "java.security.acl",
       "java.security.cert",
       "java.security.spec",
       "java.sql",
       "java.text",
       "java.util",
       "android.database.sqlite",
       "java.util.concurrent",
       "java.util.concurrent.atomic",
       "java.util.concurrent.locks",
       "java.util.jar",
       "java.util.logging",
       "java.util.prefs",
       "java.util.regex",
       "java.util.zip",
       "javax.crypto",
       "javax.crypto.spec",
       "android.graphics",
```

"javax.microedition.khronos.egl",

"javax.net",

"javax.net.ssl",

"javax.security.auth",

"javax.security.auth.callback",

"javax.security.auth.login",

"javax.security.auth.x500",

"javax.security.cert",

"javax.sql",

"javax.xml",

"javax.xml.parsers",

"junit.framework",

"junit.runner",

"org.apache.http",

"org.apache.http.auth",

"org.apache.http.auth.params",

"org.apache.http.client",

"org.apache.http.client.entity",

"org.apache.http.client.methods",

"org.apache.http.client.params",

"org.apache.http.client.protocol",

"org.apache.http.client.utils",

"org.apache.http.conn",

"org.apache.http.conn.params",

"org.apache.http.conn.routing",

"org.apache.http.conn.scheme",

"org.apache.http.conn.ssl",

"org.apache.http.conn.util",

"org.apache.http.cookie",

"org.apache.http.cookie.params",

"org.apache.http.entity",

"org.apache.http.impl",

"org.apache.http.impl.auth",

"org.apache.http.impl.client",

"org.apache.http.impl.conn",

"org.apache.http.impl.conn.tsccm",

"org.apache.http.impl.cookie",

"org.apache.http.impl.entity",

"org.apache.http.impl.io",

"org.apache.http.message",

"org.apache.http.params",

"android.app",

"org.apache.http.protocol",

"org.apache.http.util",

"org.json",

"org.w3c.dom",

"org.xml.sax",

"org.xml.sax.ext",

"org.xml.sax.helpers",

"org.xmlpull.v1",

"org.xmlpull.v1.sax2",

"android.graphics.drawable",

"android.graphics.drawable.shapes",

"android.hardware",

"android.location",

"android.media",

"android.net",

"android.net.http",

"android.net.wifi",

"android.opengl",

"android.os",

"android.preference",

"android.provider",

"android.sax",

"android.telephony",

"android.telephony.gsm",

"android.test",

"android.test.mock",

"android.test.suitebuilder",

"android.test.suitebuilder.annotation",

"android.text",

"android.text.method",

"android.text.style",

"android.content",

"android.text.util",

"android.util",

"android.view",

"android.view.animation",

"android.webkit",

"android.widget",

"dalvik.annotation",

"dalvik.system",

"java.awt.font",

"java.io",

"android.content.pm",

"java.lang",

"java.lang.annotation",

"java.lang.ref",

"java.lang.reflect",

"java.math",

"java.net",

"java.nio",

"com.android.internal.util",

"dalvik.bytecode",

"java.security.interfaces",

"javax.crypto.interfaces",

"javax.microedition.khronos.opengles",

"org.apache.commons.logging",

"org.apache.http.io",

"android.inputmethodservice",

"android.speech",

"android.text.format",

"android.appwidget",

"android.view.inputmethod",

"java.beans",

"android.gesture",

"android.accessibilityservice",

"android.speech.tts",

"android.view.accessibility",

"android.accounts",

"android.telephony.cdma",

"android.bluetooth",

"android.service.wallpaper",

"javax.xml.datatype",

"javax.xml.namespace",

"javax.xml.transform",

"javax.xml.transform.dom",

"javax.xml.transform.sax",

"javax.xml.transform.stream",

"javax.xml.validation",

"javax.xml.xpath",

"org.w3c.dom.ls",

"android.app.admin",

"android.app.backup",

"android.media.audiofx",

"android.net.sip",

"android.nfc",

"android.nfc.tech",

"android.os.storage",

"android.drm",

"android.animation",

"android.renderscript",

"android.hardware.usb",

"android.mtp",

"android.net.rtp",

"android.media.effect",

"android.net.wifi.p2p",

"android.security",

"android.service.textservice",

"android.view.textservice",

"android.hardware.input",

"android.net.nsd",

"android.net.wifi.p2p.nsd",

"android.hardware.display",

"android.service.dreams",

"android.hardware.location",

"android.service.notification",

"android.graphics.pdf",

"android.nfc.cardemulation",

"android.print",

"android.print.pdf",

"android.printservice",

"android.transition",

"android.app.job",

"android.app.usage",

"android.bluetooth.le",

"android.hardware.camera2",

"android.hardware.camera2.params",

"android.media.browse",

"android.media.projection",

"android.media.session",

"android.media.tv",

"android.service.media",

"android.service.restrictions",

"android.service.voice",

"android.system",

"android.telecom",

"android.service.carrier",

"android.app.assist",

"android.hardware.fingerprint",

"android.media.midi",

"android.security.keystore",

"android.service.chooser")

```
# method change analysis


change_unique <- data.frame(matrix(ncol = 22, nrow = 200))

change_multiple <- data.frame(matrix(ncol = 22, nrow = 200))

change <- data.frame(matrix(ncol = 22, nrow = 200))


for (j in 1:200)

{

 p_unique <- m_unique[m_unique$package == p[j],]

 p_multiple <- m_multiple_updated[m_multiple_updated$package == p[j],]


 for (i in 1:22)

 {

    p_change_unique <- merge(x = p_unique[p_unique$version==i,],y = p_unique[p_unique$version==i+1,], by =
c("class","method"),all = FALSE)

    p_change_unique$change <- p_change_unique$m_struct.x == p_change_unique$m_struct.y

    change_unique[j,i] <- nrow(p_change_unique[p_change_unique$change == FALSE,])


    p_change_multiple <- merge(x = p_multiple[p_multiple$version==i,],y = p_multiple[p_multiple$version==i+1,], by =
c("class","method"),all = FALSE)

    p_change_multiple$change <- p_change_multiple$m_struct_c.x == p_change_multiple$m_struct_c.y

    change_multiple[j,i] <- nrow(p_change_multiple[p_change_multiple$change == FALSE,])


    change <- change_unique + change_multiple


 }


}


write.csv(change,file='change.csv')
```

```
# method change for API level 13-14

android_1314 <- read.csv("13_14_input.csv", as.is = TRUE)

android_1314 <- android_1314 %>% group_by(class,package,method,version) %>% mutate(length(m_struct))

names(android_1314)[7] <- "count"

m_unique_1314 <- android_1314[android_1314$count==1,]

m_multiple_1314 <- android_1314[android_1314$count > 1,]

m_multiple_1314group <- group_by(m_multiple_1314,class,package,method,version)

m_multiple_1314updated <- summarize(m_multiple_1314group,m_struct_c = paste(m_struct, collapse = " "))

change_unique1314 <- data.frame(matrix(ncol = 1, nrow = 200))

change_multiple1314 <- data.frame(matrix(ncol = 1, nrow = 200))

change1314 <- data.frame(matrix(ncol = 1, nrow = 200))

for (j in 1:200)

{

  p_unique1314 <- m_unique_1314[m_unique_1314$package == p[j],]

  p_multiple1314 <- m_multiple_1314updated[m_multiple_1314updated$package == p[j],]

  p_change_unique1314 <- merge(x = p_unique1314[p_unique1314$version==13,],y = p_unique1314[p_unique1314$version==14,],
by = c("class","method"),all = FALSE)

  p_change_unique1314$change <- p_change_unique1314$m_struct.x == p_change_unique1314$m_struct.y

  change_unique1314[j,1] <- nrow(p_change_unique1314[p_change_unique1314$change == FALSE,])

  p_change_multiple1314 <- merge(x = p_multiple1314[p_multiple1314$version==13,],y =
p_multiple1314[p_multiple1314$version==14,], by = c("class","method"),all = FALSE)

  p_change_multiple1314$change <- p_change_multiple1314$m_struct_c.x == p_change_multiple1314$m_struct_c.y
```

```
 change_multiple1314[j,1] <- nrow(p_change_multiple1314[p_change_multiple1314$change == FALSE,])

 change1314 <- change_unique1314 + change_multiple1314

}

write.csv(change1314,file='change13_14.csv')



#method added and removed

android_maturity <- distinct(select(android_stability,package, class, method, version))

method_add <- data.frame(matrix(ncol = 22, nrow = 200))
method_remove <- data.frame(matrix(ncol = 22, nrow = 200))

for (j in 1:200)
{
 p_analysis <- android_maturity[android_maturity$package == p[j],]

 for (i in 1:22)
 {
  p_merge <- merge(x = p_analysis[p_analysis$version==i,],y = p_analysis[p_analysis$version==i+1,], by =
c("package","class","method"),all = TRUE)
   method_add[j,i] <- nrow(p_merge[is.na(p_merge$version.x),])
   method_remove[j,i] <- nrow(p_merge[is.na(p_merge$version.y),])

 }

}
```

```
write.csv(method_add,file='method_add.csv')

write.csv(method_remove,file='method_remove.csv')


write.csv(android_maturity,file = 'android_maturity.csv')


# method size for all API levels


method_size <- data.frame(matrix(ncol = 23, nrow = 200))


for (j in 1:200)

{

 p_size <- android_maturity[android_maturity$package == p[j],]


 for (i in 1:23)

 {

  method_size[j,i] <- nrow(p_size[p_size$version==i,])


 }


}


write.csv(method_size,file='method_size.csv')


# get the list of changed methods over all versions in the coplies APIs


setwd("C:/Users/mshen/Box Sync/Oracle-Google/workstreams/34_API stability analysis/android stability")


android_stability <- read.csv("android stability input.csv", as.is = TRUE)
```

```
android_stability <- android_stability %>% group_by(class,package,method,version) %>% mutate(length(m_struct))

names(android_stability)[8] <- "count"


m_unique <- android_stability[android_stability$count==1,]

m_multiple <- android_stability[android_stability$count > 1,]


m_multiple_group <- group_by(m_multiple,class,package,method,version)

m_multiple_updated <- summarize(m_multiple_group,m_struct_c = paste(m_struct, collapse = " "))


p_copied <- c("javax.sql",

              "java.beans",

              "java.lang.ref",

              "java.net",

              "java.util.logging",

              "java.util",

              "java.io",

              "java.lang",

              "java.lang.annotation",

              "java.nio",

              "java.nio.channels",

              "java.nio.channels.spi",

              "java.nio.charset",

              "java.security",

              "java.security.acl",

              "java.security.cert",

              "java.security.interfaces",

              "java.sql",

              "java.text",

              "java.util.jar",
```

```
            "java.util.prefs",
            "java.util.zip",
            "javax.crypto",
            "javax.crypto.interfaces",
            "javax.crypto.spec",
            "javax.net",
            "javax.security.auth",
            "javax.security.auth.callback",
            "javax.security.auth.login",
            "javax.security.cert",
            "java.nio.charset.spi",
            "java.security.spec",
            "java.util.regex",
            "javax.net.ssl",
            "javax.security.auth.x500",
            "java.lang.reflect",
            "java.awt.font")

# get the method change list between each API level

p_change_1 <- data.frame(class = character(),method = character(),c_struct_pre = character(),
                         m_struct_pre = character(),m_var_pre = character(),version_pre = numeric(),
                         package_pre = character(),count_pre=numeric(),c_struct_post = character(),
                         m_struct_post = character(),m_var_post = character(),version_post = numeric(),
                         package_post = character(),count_post = numeric(),change = logical())
p_change_2 <- data.frame(class = character(),method = character(),package_pre = character(),version_pre = numeric(),
                         m_struct_pre = character(),package_post = character(),version_post = numeric(),
                         m_struct_post = character(),change = logical())
```

```
for (j in 1:37)

{

  p_unique <- m_unique[m_unique$package ==p_copied[j],]

  p_multiple <- m_multiple_updated[m_multiple_updated$package == p_copied[j],]



  for (i in 1:22)

  {

    p_change_unique <- merge(x = p_unique[p_unique$version==i,],y = p_unique[p_unique$version==i+1,], by =
c("class","method"),all = FALSE)

    p_change_unique$change <- p_change_unique$m_struct.x == p_change_unique$m_struct.y

    p_change_unique <- p_change_unique[p_change_unique$change == FALSE,]

    p_change_1 <- rbind(p_change_1,p_change_unique)



    p_change_multiple <- merge(x = p_multiple[p_multiple$version==i,],y = p_multiple[p_multiple$version==i+1,], by =
c("class","method"),all = FALSE)

    p_change_multiple$change <- p_change_multiple$m_struct_c.x == p_change_multiple$m_struct_c.y

    p_change_multiple <- p_change_multiple[p_change_multiple$change == FALSE,]

    p_change_2 <- rbind(p_change_2,p_change_multiple)



  }



}


write.csv(p_change_1,file='p_change_1.csv')

write.csv(p_change_2,file='p_change_2.csv')


# get the list of change from API level 13 to 14


android_1314 <- read.csv("13_14_input.csv", as.is = TRUE)

android_1314 <- android_1314 %>% group_by(class,package,method,version) %>% mutate(length(m_struct))
```

```
names(android_1314)[7] <- "count"


m_unique_1314 <- android_1314[android_1314$count==1,]

m_multiple_1314 <- android_1314[android_1314$count > 1,]


m_multiple_1314group <- group_by(m_multiple_1314,class,package,method,version)

m_multiple_1314updated <- summarize(m_multiple_1314group,m_struct_c = paste(m_struct, collapse = " "))


p_change_1_1314 <- data.frame(class = character(),method = character(),c_struct_pre = character(),

                    m_struct_pre = character(),m_var_pre = character(),version_pre = numeric(),

                    package_pre = character(),count_pre=numeric(),c_struct_post = character(),

                    m_struct_post = character(),m_var_post = character(),version_post = numeric(),

                    package_post = character(),count_post = numeric(),change = logical())
p_change_2_1314 <- data.frame(class = character(),method = character(),package_pre = character(),version_pre = numeric(),

                    m_struct_pre = character(),package_post = character(),version_post = numeric(),

                    m_struct_post = character(),change = logical())



for (j in 1:37)

{

  p_unique1314 <- m_unique_1314[m_unique_1314$package == p_copied[j],]

  p_multiple1314 <- m_multiple_1314updated[m_multiple_1314updated$package == p_copied[j],]


  p_change_unique1314 <- merge(x = p_unique1314[p_unique1314$version==13,],y = p_unique1314[p_unique1314$version==14,],
by = c("class","method"),all = FALSE)

  p_change_unique1314$change <- p_change_unique1314$m_struct.x == p_change_unique1314$m_struct.y

  p_change_unique1314 <- p_change_unique1314[p_change_unique1314$change == FALSE,]

  p_change_1_1314 <- rbind(p_change_1_1314,p_change_unique1314)
```

```
 p_change_multiple1314 <- merge(x = p_multiple1314[p_multiple1314$version==13,],y =
p_multiple1314[p_multiple1314$version==14,], by = c("class","method"),all = FALSE)

 p_change_multiple1314$change <- p_change_multiple1314$m_struct_c.x == p_change_multiple1314$m_struct_c.y

 p_change_multiple1314 <- p_change_multiple1314[p_change_multiple1314$change == FALSE,]

 p_change_2_1314 <- rbind(p_change_2_1314,p_change_multiple1314)


}


write.csv(p_change_1_1314,file='p_change_1_1314.csv')

write.csv(p_change_2_1314,file='p_change_2_1314.csv')
```

R Script for Java Method Changes:

```
install.packages("dplyr")

library(dplyr)


m_unique <- java_stability[java_stability$count==1,]

m_multiple <- java_stability[java_stability$count > 1,]


#m_multiple$m_str_var <- paste(m_multiple$m_struct,m_multiple$m_var)


#write.csv(m_multiple,file='m_multiple.csv')


m_multiple_group <- group_by(m_multiple,class,package,method,version)

m_multiple_updated <- summarize(m_multiple_group,m_struct_c = paste(m_struct, collapse = " "))


#android_stability$m_str_var <- paste(android_stability$m_struct,android_stability$m_var)

#android_stability$m_char <- nchar(android_stability$m_str_var)
```

```
p <- c("java.applet",
       "java.awt",
       "java.awt.image",
       "java.awt.peer",
       "java.io",
       "java.lang",
       "java.net",
       "java.util",
       "java.awt.datatransfer",
       "java.awt.event",
       "java.beans",
       "java.lang.reflect",
       "java.math",
       "java.rmi",
       "java.rmi.dgc",
       "java.rmi.registry",
       "java.rmi.server",
       "java.security",
       "java.security.acl",
       "java.security.interfaces",
       "java.sql",
       "java.text",
       "java.text.resources",
       "java.util.zip",
       "java.awt.color",
       "java.awt.dnd",
       "java.awt.dnd.peer",
       "java.awt.font",
       "java.awt.geom",
```

"java.awt.im",

"java.awt.image.renderable",

"java.awt.print",

"java.beans.beancontext",

"java.lang.ref",

"java.rmi.activation",

"java.security.cert",

"java.security.spec",

"java.util.jar",

"javax.accessibility",

"javax.swing",

"javax.swing.border",

"javax.swing.colorchooser",

"javax.swing.event",

"javax.swing.filechooser",

"javax.swing.plaf",

"javax.swing.plaf.basic",

"javax.swing.plaf.metal",

"javax.swing.plaf.multi",

"javax.swing.table",

"javax.swing.text",

"javax.swing.text.html",

"javax.swing.text.html.parser",

"javax.swing.text.rtf",

"javax.swing.tree",

"javax.swing.undo",

"org.omg.CORBA",

"org.omg.CORBA.DynAnyPackage",

"org.omg.CORBA.ORBPackage",

"org.omg.CORBA.portable",

"org.omg.CORBA.TypeCodePackage",

"org.omg.CosNaming",

"org.omg.CosNaming.NamingContextPackage",

"java.awt.im.spi",

"javax.naming",

"javax.naming.directory",

"javax.naming.event",

"javax.naming.ldap",

"javax.naming.spi",

"org.omg.CORBA_2_3",

"org.omg.CORBA_2_3.portable",

"org.omg.SendingContext",

"org.omg.stub.java.rmi",

"java.nio",

"java.nio.channels",

"java.nio.channels.spi",

"java.nio.charset",

"java.nio.charset.spi",

"java.util.logging",

"java.util.prefs",

"java.util.regex",

"javax.imageio",

"javax.imageio.event",

"javax.imageio.metadata",

"javax.imageio.plugins.jpeg",

"javax.imageio.spi",

"javax.imageio.stream",

"javax.print",

"javax.print.attribute",

"javax.print.attribute.standard",

"javax.print.event",

"javax.rmi",

"javax.rmi.CORBA",

"javax.security.auth",

"javax.security.auth.callback",

"javax.security.auth.kerberos",

"javax.security.auth.login",

"javax.security.auth.spi",

"javax.security.auth.x500",

"javax.sql",

"javax.xml.parsers",

"javax.xml.transform",

"javax.xml.transform.dom",

"javax.xml.transform.sax",

"javax.xml.transform.stream",

"org.apache.crimson.jaxp",

"org.apache.crimson.parser",

"org.apache.crimson.tree",

"org.apache.crimson.util",

"org.apache.xalan.client",

"org.apache.xalan.extensions",

"org.apache.xalan.lib",

"org.apache.xalan.lib.sql",

"org.apache.xalan.processor",

"org.apache.xalan.res",

"org.apache.xalan.serialize",

"org.apache.xalan.templates",

"org.apache.xalan.trace",

"org.apache.xalan.transformer",

"org.apache.xalan.xslt",

"org.apache.xml.dtm",

"org.apache.xml.dtm.ref",

"org.apache.xml.dtm.ref.dom2dtm",

"org.apache.xml.dtm.ref.sax2dtm",

"org.apache.xml.utils",

"org.apache.xml.utils.res",

"org.apache.xml.utils.synthetic",

"org.apache.xml.utils.synthetic.reflection",

"org.apache.xpath",

"org.apache.xpath.axes",

"org.apache.xpath.compiler",

"org.apache.xpath.functions",

"org.apache.xpath.objects",

"org.apache.xpath.operations",

"org.apache.xpath.patterns",

"org.apache.xpath.res",

"org.ietf.jgss",

"org.omg.CosNaming.NamingContextExtPackage",

"org.omg.Dynamic",

"org.omg.DynamicAny",

"org.omg.DynamicAny.DynAnyFactoryPackage",

"org.omg.DynamicAny.DynAnyPackage",

"org.omg.IOP",

"org.omg.IOP.CodecFactoryPackage",

"org.omg.IOP.CodecPackage",

"org.omg.Messaging",

"org.omg.PortableInterceptor",

"org.omg.PortableInterceptor.ORBInitInfoPackage",

"org.omg.PortableServer",

"org.omg.PortableServer.CurrentPackage",

"org.omg.PortableServer.POAManagerPackage",

"org.omg.PortableServer.POAPackage",

"org.omg.PortableServer.portable",

"org.omg.PortableServer.ServantLocatorPackage",

"org.w3c.dom",

"org.w3c.dom.css",

"org.w3c.dom.events",

"org.w3c.dom.html",

"org.w3c.dom.stylesheets",

"org.w3c.dom.traversal",

"org.w3c.dom.views",

"org.xml.sax",

"org.xml.sax.ext",

"org.xml.sax.helpers",

"java.lang.annotation",

"java.lang.instrument",

"java.lang.management",

"java.util.concurrent",

"java.util.concurrent.atomic",

"java.util.concurrent.locks",

"javax.imageio.plugins.bmp",

"javax.management",

"javax.management.loading",

"javax.management.modelmbean",

"javax.management.monitor",

"javax.management.openmbean",

"javax.management.relation",

"javax.management.remote",

"javax.management.remote.rmi",

"javax.management.timer",

"javax.rmi.ssl",

"javax.security.sasl",

"javax.sound.midi",

"javax.sound.midi.spi",

"javax.sound.sampled",

"javax.sound.sampled.spi",

"javax.sql.rowset",

"javax.sql.rowset.serial",

"javax.sql.rowset.spi",

"javax.swing.plaf.synth",

"javax.xml",

"javax.xml.datatype",

"javax.xml.namespace",

"javax.xml.validation",

"javax.xml.xpath",

"org.w3c.dom.bootstrap",

"org.w3c.dom.ls",

"org.w3c.dom.ranges",

"java.text.spi",

"java.util.spi",

"javax.annotation",

"javax.annotation.processing",

"javax.lang.model",

"javax.lang.model.element",

"javax.lang.model.type",

"javax.lang.model.util",

"programelements",

"types",

"javax.script",

"javax.tools",

"javax.xml.bind",

"javax.xml.bind.annotation",

"javax.xml.bind.annotation.adapters",

"javax.xml.bind.attachment",

"javax.xml.bind.helpers",

"javax.xml.bind.util",

"javax.xml.crypto",

"javax.xml.crypto.dom",

"javax.xml.crypto.dsig",

"javax.xml.crypto.dsig.dom",

"javax.xml.crypto.dsig.keyinfo",

"javax.xml.crypto.dsig.spec",

"javax.xml.soap",

"javax.xml.stream",

"javax.xml.stream.events",

"javax.xml.stream.util",

"javax.xml.transform.stax",

"javax.xml.ws",

"javax.xml.ws.handler",

"javax.xml.ws.handler.soap",

"javax.xml.ws.http",

"javax.xml.ws.soap",

"javax.xml.ws.spi",

```
    "org.w3c.dom.xpath",
    "java.lang.invoke",
    "java.nio.file",
    "java.nio.file.attribute",
    "java.nio.file.spi",
    "javax.security.cert",
    "javax.swing.plaf.nimbus",
    "javax.xml.ws.spi.http",
    "javax.xml.ws.wsaddressing",
    "java.time",
    "java.time.chrono",
    "java.time.format",
    "java.time.temporal",
    "java.time.zone",
    "java.util.function",
    "java.util.stream")

# method change analysis

change_unique <- data.frame(matrix(ncol = 8, nrow = 248))
change_multiple <- data.frame(matrix(ncol = 8, nrow = 248))
change <- data.frame(matrix(ncol = 8, nrow = 248))

for (j in 1:248)
{
  p_unique <- m_unique[m_unique$package == p[j],]
  p_multiple <- m_multiple_updated[m_multiple_updated$package == p[j],]

  for (i in 1:8)
```

```
  {

    p_change_unique <- merge(x = p_unique[p_unique$version==i,],y = p_unique[p_unique$version==i+1,], by =
c("class","method"),all = FALSE)

    p_change_unique$change <- p_change_unique$m_struct.x == p_change_unique$m_struct.y

    change_unique[j,i] <- nrow(p_change_unique[p_change_unique$change == FALSE,])



    p_change_multiple <- merge(x = p_multiple[p_multiple$version==i,],y = p_multiple[p_multiple$version==i+1,], by =
c("class","method"),all = FALSE)

    p_change_multiple$change <- p_change_multiple$m_struct_c.x == p_change_multiple$m_struct_c.y

    change_multiple[j,i] <- nrow(p_change_multiple[p_change_multiple$change == FALSE,])



    change <- change_unique + change_multiple


  }


}


write.csv(change,file='change.csv')




#method added and removed


java_maturity <- distinct(select(java_stability,package, class, method, version))


method_add <- data.frame(matrix(ncol = 8, nrow = 248))

method_remove <- data.frame(matrix(ncol = 8, nrow = 248))


for (j in 1:248)

{

  p_analysis <- java_maturity[java_maturity$package == p[j],]
```

```
 for (i in 1:8)

 {

  p_merge <- merge(x = p_analysis[p_analysis$version==i,],y = p_analysis[p_analysis$version==i+1,], by =
c("package","class","method"),all = TRUE)

  method_add[j,i] <- nrow(p_merge[is.na(p_merge$version.x),])

  method_remove[j,i] <- nrow(p_merge[is.na(p_merge$version.y),])


 }


}


write.csv(method_add,file='method_add.csv')

write.csv(method_remove,file='method_remove.csv')


#write.csv(android_maturity,file = 'android_maturity.csv')


# method size for all API levels


method_size <- data.frame(matrix(ncol = 9, nrow = 248))


for (j in 1:248)

{

 p_size <- java_maturity[java_maturity$package == p[j],]


 for (i in 1:9)

 {

  method_size[j,i] <- nrow(p_size[p_size$version==i,])


 }
```

```
}

write.csv(method_size,file='method_size.csv')

# 37 API size over versions

p_copied <- c("javax.sql",
              "java.beans",
              "java.lang.ref",
              "java.net",
              "java.util.logging",
              "java.util",
              "java.io",
              "java.lang",
              "java.lang.annotation",
              "java.nio",
              "java.nio.channels",
              "java.nio.channels.spi",
              "java.nio.charset",
              "java.security",
              "java.security.acl",
              "java.security.cert",
              "java.security.interfaces",
              "java.sql",
              "java.text",
              "java.util.jar",
              "java.util.prefs",
              "java.util.zip",
```

"javax.crypto",

"javax.crypto.interfaces",

"javax.crypto.spec",

"javax.net",

"javax.security.auth",

"javax.security.auth.callback",

"javax.security.auth.login",

"javax.security.cert",

"java.nio.charset.spi",

"java.security.spec",

"java.util.regex",

"javax.net.ssl",

"javax.security.auth.x500",

"java.lang.reflect",

"java.awt.font")

APPENDIX R – Fragmentation Chart

Packages In Java SE 5 - 37 Copied Java Packages In Green

java.applet	java.awt.image.renderable	java.math	java.rmi.server	java.util.concurrent.locks	javax.imageio	javax.management.openmbean	javax.net.ssl	javax.security.auth.login	javax.sql.rowset.serial
java.awt	java.awt.print	java.net	java.security	java.util.jar	javax.imageio.event	javax.management.relation	javax.print	javax.security.auth.spi	javax.sql.rowset.spi
java.awt.color	java.beans	java.nio	java.security.acl	java.util.logging	javax.imageio.metadata	javax.management.remote	javax.print.attribute	javax.security.auth.x500	javax.swing
java.awt.datatransfer	java.beans.beancontext	java.nio.channels	java.security.cert	java.util.prefs	javax.imageio.plugins.bmp	javax.management.remote.rmi	javax.print.attribute.standard	javax.security.cert	javax.swing.border
java.awt.dnd	java.io	java.nio.channels.spi	java.security.interfaces	java.util.regex	javax.imageio.plugins.jpeg	javax.management.timer	javax.print.event	javax.security.sasl	javax.swing.colorchooser
java.awt.event	java.lang	java.nio.charset	java.security.spec	java.util.zip	javax.imageio.spi	javax.naming	javax.rmi	javax.sound.midi	javax.swing.event
java.awt.font	java.lang.annotation	java.nio.charset.spi	java.sql	javax.accessibility	javax.imageio.stream	javax.naming.directory	javax.rmi.CORBA	javax.sound.midi.spi	javax.swing.filechooser
java.awt.geom	java.lang.instrument	java.rmi	java.text	javax.activity	javax.management	javax.naming.event	javax.rmi.ssl	javax.sound.sampled	javax.swing.plaf
java.awt.im	java.lang.management	java.rmi.activation	java.util	javax.crypto	javax.management.loading	javax.naming.ldap	javax.security.auth	javax.sound.sampled.spi	javax.swing.plaf.basic
java.awt.im.spi	java.lang.ref	java.rmi.dgc	java.util.concurrent	javax.crypto.interfaces	javax.management.modelmbean	javax.naming.spi	javax.security.auth.callback	javax.sql	javax.swing.plaf.metal
java.awt.image	java.lang.reflect	java.rmi.registry	java.util.concurrent.atomic	javax.crypto.spec	javax.management.monitor	javax.net	javax.security.auth.kerberos	javax.sql.rowset	javax.swing.plaf.multi
org.omg.PortableServer.POAPackage	org.omg.PortableServer.CurrentPackage	org.omg.PortableServer.POAManagerPackage	org.omg.PortableServer.portable	org.omg.PortableServer.ServantLocatorPackage	org.omg.SendingContext	org.omg.stub.java.rmi	org.w3c.dom	org.w3c.dom.bootstrap	org.w3c.dom.events
org.omg.Dynamic	org.omg.DynamicAny	org.xml.sax.helpers	org.xml.sax.ext	org.omg.DynamicAny.DynAnyPackage	org.omg.IOP	org.omg.IOP.CodecFactoryPackage	org.omg.IOP.CodecPackage	org.omg.Messaging	org.omg.PortableInterceptor
org.omg.PortableServer	org.omg.CosNaming.NamingContextPackage	org.omg.CORBA_2_3	org.omg.CORBA_2_3.portable	org.omg.CORBA.DynAnyPackage	org.omg.CORBA.ORBPackage	org.omg.CORBA.portable	org.omg.CORBA.TypeCodePackage	org.omg.CosNaming	org.omg.DynamicAny.DynAnyFactoryPackage
javax.xml.namespace	javax.xml.parsers	javax.xml.transform	javax.xml.transform.dom	javax.xml.transform.sax	javax.xml.transform.stream	javax.xml.validation	javax.xml.xpath	org.ietf.jgss	org.omg.CORBA

javax.swing.tree	javax.swing.undo	javax.transaction	javax.transaction.xa	javax.xml	org.w3c.dom.ls	org.omg.PortableInterceptor.ORBInitInfoPackage	org.omg.CosNaming.NamingContextExtPackage	org.xml.sax	javax.xml.datatype
				javax.swing.text	javax.swing.table	javax.swing.plaf.synth	javax.swing.text.html	javax.swing.text.html.parser	javax.swing.text.rtf

APPENDIX S – Java SE and Java ME

37 Java Packages At Issue	Java SE 5.0	Java ME			
		CDC		CLDC	
		1.0a[148]	1.1.2[149]	1.0a[150]	1.1[151]
java.awt.font	x				
java.beans	x				
java.io	x	x	x	x	x
java.lang	x	x	x	x	x
java.lang.annotation	x				
java.lang.ref	x	x	x		x
java.lang.reflect	x	x	x		
java.net	x		x		
java.nio	x				
java.nio.channels	x				
java.nio.channels.spi	x				
java.nio.charset	x				
java.nio.charset.spi	x				
java.security	x	x	x		
java.security.acl	x				
java.security.cert	x	x	x		
java.security.interfaces	x				
java.security.spec	x				
java.sql	x				
java.text	x	x	x		
java.util	x	x	x	x	x
java.util.jar	x	x	x		
java.util.logging	x				
java.util.prefs	x				
java.util.regex	x				
java.util.zip	x	x	x		
javax.crypto	x				
javax.crypto.interfaces	x				
javax.crypto.spec	x				
javax.net	x				
javax.net.ssl	x				
javax.security.auth	x				
javax.security.auth.callback	x				
javax.security.auth.login	x				
javax.security.auth.x500	x				
javax.security.cert	x				
javax.sql	x				
Java ME Specific Packages					
javax.microedition.io		x	x	x	x

[148] Java ME, CDC 1.0.2 is defined in JSR-36. *See* http://download.oracle.com/otn-pub/jcp/7825-j2me_cdc-1.0a-mr-spec-oth-JSpec/j2me_cdc-1_0a-mr-spec.zip.
[149] Java ME, CDC 1.1.2 is defined in JSR-218. *See* http://download.oracle.com/otn-pub/jcp/cdc-1.1-fr-eval-oth-spec-JSpec/cdc-1_1-fr-spec.zip; *see also* http://docs.oracle.com/javame/config/cdc/ref-impl/cdc1.1.2/jsr218/
[150] Java ME, CLDC 1.0 is defined in JSR-30. *See* http://download.oracle.com/otn-pub/jcp/7587-j2me_cldc-1.0a-fr-spec-oth-JSpec/cldc-1_0a-spec.zip; *see also* https://docs.oracle.com/javame/config/cldc/ref-impl/cldc1.0/cldcapi.pdf
[151] Java ME, CLDC 1.1 is defined in JSR-139. *See* http://download.oracle.com/otn-pub/jcp/7247-j2me_cldc-1.1-fr-spec-oth-JSpec/cldc-1_1-fr-spec.zip; *see also* http://docs.oracle.com/javame/config/cldc/ref-impl/cldc1.1/jsr139/

PROOF OF SERVICE BY KITEWORKS

I, José E. Valdés, am over the age of eighteen years old and not a party to the within-entitled action. My place of employment and business address is Orrick, Herrington & Sutcliffe LLP, 1000 Marsh Road, Menlo Park, California 94025.

On January 8, 2016, I served the following documents:

EXPERT REPORT OF DR. CHRIS F. KEMERER.

on the interested parties in this action by electronic service [Fed. Rule Civ. Proc. 5(b)] by electronically mailing a true and correct copy, pursuant to the parties agreement, to the following email addresses:

DALVIK-KVN@kvn.com
JCooper@fbm.com
gglas@fbm.com

I declare under penalty of perjury under the laws of the State of California and the United States that the foregoing is true and correct.

Executed on January 8, 2016, at San Francisco, California.

/s/ José E. Valdés
José E. Valdés